TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Study, design and construction of a Bi-Rotor and Tri-Rotor MAV platforms.

TITULACIÓ: Enginyeria Tècnica Aeronàutica, especialitat Aeronavegació

AUTORS: Eric Arnal
Carlos Escusa

DIRECTOR: Oscar Casas Piedrafita

DATA: 16 de juliol de 2009
Overview

The aim of this project is to create two Micro Air Vehicles (MAVs) platforms.

This includes a previous study of MAVs characteristics and its background, the design of the vehicles and all of their parts, the creation of them, their mechanical assembling and specific hardware and software development for each vehicle.

The result is two platforms completely ready to program, so future users and researchers can focus their work on the application software, leaving all the hardware apart.
INDEX

CHAPTER 1. INTRODUCTION ................................................................................. 1

1.1 Background ........................................................................................................ 1

1.2 Objectives ........................................................................................................... 10

1.3 General MAV architecture .................................................................................. 12
  1.3.1 Airframe ....................................................................................................... 12
  1.3.2 Communication ............................................................................................. 13
  1.3.3 Processor ....................................................................................................... 13
  1.3.4 Sensors .......................................................................................................... 13
  1.3.5 Power ........................................................................................................... 13

CHAPTER 2. DESIGN AND DYNAMICS .............................................................. 14

2.1 Structure .............................................................................................................. 14
  2.1.1 Bi-Rotor Structure .......................................................................................... 14
  2.1.2 Tri-Rotor Structure ....................................................................................... 19

2.2 Dynamics ............................................................................................................. 22
  2.2.1 Bi-Rotor ......................................................................................................... 23
  2.2.2 Translation ..................................................................................................... 24
  2.2.3 Rotation ......................................................................................................... 29
  2.2.4 Tri-Rotor ....................................................................................................... 33

2.3 Basics of control ................................................................................................. 37

CHAPTER 3. HARDWARE .................................................................................. 41

3.1 Actuators .............................................................................................................. 41
  3.1.1 Motors .......................................................................................................... 41
  3.1.2 Servos .......................................................................................................... 44

3.2 Motors & Blades Study ..................................................................................... 46
  3.2.1 Motor conclusions ......................................................................................... 51
  3.2.2 Available payloads ....................................................................................... 52

3.3 RF Data-Link ..................................................................................................... 52
  3.3.1 Transmitter ................................................................................................... 52
  3.3.2 Receiver ........................................................................................................ 54

3.4 Power system .................................................................................................... 55

3.5 Microcontroller and boards .............................................................................. 60
  3.5.1 FlyShield ...................................................................................................... 61
  3.5.2 PowerShield .................................................................................................. 67

CHAPTER 4. SOFTWARE ................................................................................ 69

4.1 AVOIDANCE SYSTEM ..................................................................................... 69
  4.1.1 Objective ...................................................................................................... 69
  4.1.2 Parts ............................................................................................................. 70
  4.1.3 Code ............................................................................................................. 70
4.2 GROUND CONTACT SYSTEM ........................................................................................ 72
  4.2.1 Objective ............................................................................................................... 72
  4.2.2 Parts ..................................................................................................................... 72

4.3 LIGHTNING SYSTEM ................................................................................................. 73
  4.3.1 Objective ............................................................................................................... 73
  4.3.2 Parts ..................................................................................................................... 73
  4.3.3 Code ..................................................................................................................... 76

CHAPTER 5. CONCLUSIONS AND FUTURE PERSPECTIVES ............. 78

REFERENCES .............................................................................................................. 79

ANNEX 1: MSP430 Software .................................................................................. 81

ANNEX 2: Weights ................................................................................................. 86

ANNEX 3: Plans ......................................................................................................... 87
FIGURES INDEX

Fig. 1.1 Flying bird .............................................................................................. 1
Fig. 1.3 Aerial carriage (Credit, Hiller Aviation Museum [3])............................... 2
Fig. 1.2 Leonardo Da Vinci’s air screw (Credit, Hiller Aviation Museum [3])...... 2
Fig. 1.4 Ponton d’Amecourt’s helicopters (Credit, Hiller Aviation Museum [3])... 3
Fig. 1.5 Cornu two-rotor flying machine (Credit, Hiller Aviation Museum [3]) .... 3
Fig. 1.6 Boris Yur’ev’s aircraft (Credit [3] [5]) ...................................................... 4
Fig. 1.7 Petroczy – Von Karman’s flying machine (Credit, Hiller Aviation Museum [3]) ................................................................................................ 4
Fig. 1.8 Stringfellow’s model of 1868 (Credit [6]) ................................................ 5
Fig. 1.9 Micro Air Vehicle Flight Regime Compared to Existing Flight Vehicles. 6
Fig. 1.10 Four-Rotor rotorcraft and X6 Tri-rotor from Draganflyer Innovations Inc (Credits [8]). ................................................................................................ 7
Fig. 1.11 Aerosonde Robotic Aircraft (Credits [9]) .............................................. 8
Fig. 1.12 Mikado aircraft - EMT, Germany (Credits [10]).................................... 8
Fig. 1.13 Emergency wireless network established by a swarm of MAVs [12]… 9
Fig. 1.14 Artistic example of a swarm of MAVs used in an airport [12].............. 9
Fig. 1.15 Our Bi-Rotor basic idea ..................................................................... 11
Fig. 1.16 Our Tri-Rotor basic idea ..................................................................... 11
Fig. 1.17 General MAV diagram ..................................................................... 12
Fig. 2.1 Bi-rotor scheme ................................................................................... 14
Fig. 2.2 Carbon fiber rods ............................................................................... 15
Fig. 2.3 Steel rods ............................................................................................ 15
Fig. 2.4 Moving arm assembling ....................................................................... 16
Fig. 2.5 Arm - servo union ............................................................................... 16
Fig. 2.6 Body assembling .................................................................................. 16
Fig. 2.7 Force transmission point ..................................................................... 17
Fig. 2.8 Arm separating avoidance ................................................................... 17
Fig. 2.9 Bi-Rotor design .................................................................................... 18
Fig. 2.10 Bi-Rotor photo ................................................................................... 18
Fig. 2.11 Tri-Rotor ............................................................................................ 19
Fig. 2.12 Tri-Rotor CAD parts .......................................................................... 20
Fig. 2.13 Draganflyer motor mount and vertical riser leg .................................. 20
Fig. 2.14 Structure central brass piece ............................................................. 21
Fig. 2.15 Closing the structure ........................................................................ 21
Fig. 2.16 Tri-Rotor design ................................................................................ 22
Fig. 2.17 Tri-Rotor photo .................................................................................. 22
Fig. 2.18 Basic Force diagram .......................................................................... 23
Fig. 2.19 Vertical axis force scheme ................................................................. 24
Fig. 2.20 Momentums ...................................................................................... 27
Fig. 2.21 β≠0 with symmetrical arms movement ............................................ 28
Fig. 2.22 β ≠ 0 with non-symmetrical arms movement ..................................... 29
Fig. 2.23 Accelerated circular movement .......................................................... 29
Fig. 2.24 Model diagram .................................................................................. 30
Fig. 2.25 Tri-Rotor arm ..................................................................................... 33
Fig. 2.26 Lift force produced by both motors .................................................. 33
Fig. 2.27 Tri-Rotor dynamics .......................................................................... 34
Fig. 2.28 Tri-Rotor translation movement ........................................................ 34
TABLES INDEX

Table 2.1 PID effect ................................................................. 38
Table 3.1. Brushed and brushless motors comparison .................... 44
Table 3.2 Bi-Rotor and Tri-Rotor performances ............................ 52
Table 3.3 Battery performances comparison .................................. 56
Table 3.4 Bi-rotor consumptions and capacity required .................. 57
Table 3.5. VTEC lithium polymer battery specifications .................. 58
Table 3.6. Tri-rotor consumptions and capacity required ................. 59
Table 3.7. VTEC lithium polymer battery specifications .................. 59
Table 3.8 Accelerometer output voltage values with $V_{cc} = 3.3 V$ .... 63
CHAPTER 1. INTRODUCTION

1.1 Background

Unmanned aerial vehicles (UAVs) are crafts capable of flight without an onboard pilot. They can be controlled remotely by an operator, or autonomously via preprogrammed flight paths. Recently they have reached unprecedented levels of growth in diverse military and civilian application domains. In modern times, UAVs appeared during the World War I (1917). However, the idea for a ‘flying machine’ originated and it was first conceived about 2,500 ago.

It has been documented that the first major breakthrough contribution to autonomous flying mechanisms occurred during the era of Pythagoras. The first breakthrough is attributed to Archytas from the city of Tarantas in South Italy, known as Archytas the Tarantine, also referred to as Leonardo Da Vinci of the Ancient World. Archytas was not only the inventor of the number ‘one’, ‘the father of ‘1’ in number theory, but he was also the first engineer. By applying a series of geometric notions and observations to the study of structures, links and joints, he created Mechanics. In 425 B.C. he created the first UAV of all times by building a mechanical bird, a pigeon that could fly by moving its wings getting energy from a mechanism in its stomach. It is alleged that it flew about 200 meters before falling to the ground, once all energy was used [1].

During the same era of the Pythagorean Mathematicians, at another part of the Ancient World, in China, at about 400 B.C., the Chinese were the first to document the idea of a vertical flight aircraft. The earliest version of the Chinese top consisted of feathers at the end of a stick. The stick was spun between the hands to generate enough lift before released into free flight.

More than seventeen centuries later, the initial idea attributed to Archytas surfaced again: a similar ‘flying bird’, credited to some unknown engineer of the Renaissance was documented. It is not known whether this new design was based on Archytas’ idea; however, the concept was very similar (Figure 1.1).

Leonardo Da Vinci, in 1483, designed an aircraft capable of hovering, called aerial screw or air gyroscope, shown in Figure 1.2. It had a 5 meter diameter and the idea was to make the shaft turn and if enough force were applied, the machine could spin and fly. This machine is considered by some experts as the ancestor of today’s helicopter [2] [3].

Fig. 1.1 Flying bird
Further, Da Vinci devised a mechanical bird in 1508 that could flap its wings by means of a double crank mechanism as it descended along a cable [4].

Two additional designs based on the initial Chinese top idea were documented in 1754 and 1783, respectively. The first is credited to Mikhail Lomonosov who designed a coaxial rotor powered by a wound-up spring device. The second is credited to Launoy and Bienvenue whose model consisted of a counter rotating set of turkey feathers [2] [3].

Figure 1.3 illustrates George Cayley’s aerial carriage that was designed in 1843; it is a convertible plane capable of hovering, which remained an idea due to the fact that the only available power plants at that time were steam engines that could not be used for powered flight [2] [3].

A vertical flight machine was also designed in the 1840’s by Horatio Phillips. A miniature boiler was used to generate steam that was ejected out of blade tips [2]. However, it was Ponton d’Amecourt in the 1860’s who flew small helicopter models powered by steam [2] [3] (Figure 1.4).
Additional vertical flying models were introduced between 1860 and 1907. The one standing out was introduced by Thomas Alva Edison who in the 1880’s experimented with different rotor configurations, eventually using an electric motor for power [2] [3]. Through his experiments it was revealed that for best hovering abilities, a large diameter rotor was needed with low blade area. In 1907, Paul Cornu developed a two-rotor vertically flying machine that presumably carried the first human off the ground for the first time (Figure 1.5). Rotors rotated in opposite directions, the machine flew for about 20 seconds and was merely lifted off the ground.

The major breakthrough of modern times in helicopter history was the Igor Ivanovitch Sikorsky helicopter, even though his first prototype built by 1909, a non-piloted coaxial helicopter, never flew because of vibration problems and lack of a powerful engine. Russia’s contribution came in 1912; Boris Yur’ev’s design included a main rotor and a tail rotor (used for the first time), see Figure 1.6, while he was the first to propose cyclic pitch for rotor control.
During World War I, military interest contributed to the advancement of these vehicles. Von Karman and Petrosczy, both from Germany, and the Hungarian Asboth intended to produce, without success, a lifting device to replace kite balloons for observation consisted of two superimposed lifting propellers (Figure 1.7).

It was not until late in World War I that major advances were made. The quality and quantity of production materials increased, and great improvements were made in the field of engine technology. With better technology and more need, the next step in advancement would soon come.

Of course, in parallel with building vertically flying machines and helicopters, fixed wing aircraft started to evolve over the last one hundred plus years, with the first flight demonstrated by the Wright brothers in 1903. Focusing on unmanned fixed-wing aircraft, major breakthroughs happened over the past thirty years.

But, in fact, we want to focus our project on small unmanned air vehicles, commonly termed micro air vehicles (MAVs). Their history really began with the development of model airplanes in the 19th century. Most aeronautical experimenters built models of their designs in order to discover if they would work. John Stringfellow and William Henson from England combined their talents in 1848 to build a steam powered propeller driven model aircraft with a 10 foot wingspan called the Aerial Steam Carriage. This model successfully flew for a distance of approximately 55 meters. Another Stringfellow model was flown on a wire guide inside the Crystal Palace of London in 1868 (Figure 1.8).
Eyewitnesses reported that the steam powered tri-winged airplane generated lift and only used the wire guide to keep from crashing into walls [6].

![Fig. 1.8 Stringfellow's model of 1868 (Credit [6])](image)

The American scientific Samuel Langley in 1896 successfully flew a steam powered model he called Aerodrome Number 5 down the Potomac River for 1.2 km [6]. The development of radio controlled model airplanes in the 20th century was also very important. The improvement in propulsion systems from rubber bands or steam power to liquid fuel internal combustions engines and later battery powered electric motors made it possible to produce longer flights. Also, the development of miniature radio receivers and control components in the 1990s also had a large impact on the ability to design a very small flying vehicle. Once the aerodynamics and control of small aircraft models with a mass less than 200 grams were better studied, the micro-air-vehicle was born.

The need to obtain better knowledge about fundamental aspects of flight for small vehicles has started a high quality research in the area of micro air vehicle. These aircraft have a set of characteristics very different from traditional aircraft due to their small Reynolds number more common to the small insects than to a plain (Figure 1.9). While naturalists have seriously studied bird and insect flight for more than half a century, our basic understanding of the aerodynamics encountered here is very limited. And if our understanding of low Reynolds number effects is limited, our ability to mechanize flight under these conditions has been even more elusive.

Nowadays, their demand is quickly increasing. It is essential to state that unmanned airplanes are basically used for military applications; however Vertical Take-Off and Landing (VTOL) applications extend to the non-military domains as well. VTOL military applications include surveillance and reconnaissance, combat uses and testing for new weapon systems. Nonmilitary applications include pipelines and power lines inspection and surveillance, border patrol, rescue missions, region surveillance, oil and natural gas search, fire prevention, topography and natural disasters, as well as agricultural applications (mostly in Japan). There has been a rising level of interest and investment in better vehicle designs in the last years. As such, in 1997, the total income of the UAV global market, including the Vertical Take-Off and Landing segment, reached $2.27 billion dollars [7]. Also miniaturized components are enabling many fast advances, like the development of micro-electromechanical
systems including: micro gas turbine engines, CCD-array cameras, infra red sensors, substance detectors and on-board computers.

![Graph showing Reynolds Number vs. Gross Weight (lbs)](image)

**Fig. 1.9 Micro Air Vehicle Flight Regime Compared to Existing Flight Vehicles**

A sample list of major accomplishments on micro aerial vehicles events and research groups that have had an impact on its development are presented below.

- **DARPA:** The US Defense Advanced Research Projects Agency (DARPA) is working with Oklahoma State University (OSU) to develop technologies for use in micro aerial vehicles (MAVs), including plasma actuators and inflatable wings. ([http://www.darpa.mil/](http://www.darpa.mil/))

- **Notre Dame Micro Aerial Vehicle Development Group:** Is comprised of graduate and undergraduate students working towards the goal of developing optimized, mission-capable micro aerial vehicles. The current focus of the MAV-DG is the design, construction, and flight-testing of vehicles to be entered in the annual Micro Aerial Vehicle Competitions. The MAV-DG utilizes knowledge gained through wind tunnel experiments and computational studies to develop practical design-optimization procedures applicable to MAV design. ([http://www.nd.edu/~mav/](http://www.nd.edu/~mav/))

- **USC MAV Program:** The USC Micro aerial Vehicle Program is a student organization of the University of Southern California committed to experimental learning. It gives engineering and science students an opportunity to apply lessons learned in the classroom to the analysis, design and fabrication of radio controlled MAV. ([http://ae-www.usc.edu/student/mav/](http://ae-www.usc.edu/student/mav/))
✓ **University of Florida Research Group:** They are actively designing experimental MAV and small UAV flight platforms and optimizing the fabrication techniques of these aircraft. ([http://mav.mae.ufl.edu/morph/index.html](http://mav.mae.ufl.edu/morph/index.html))

✓ **MAV Research at Brigham Young University:** Research efforts are being made at Brigham Young University related to the control of miniature aerial vehicles (MAVs). Recent results in the areas of vector field path following, precision landing and target prosecution, target localization, obstacle detection and avoidance, tailsitter aircraft control, and cooperative control can be appreciated. ([http://www.me.byu.edu/mav/](http://www.me.byu.edu/mav/))

✓ **Technical Committee on Aerial Robotics and Unmanned Aerial Vehicles:** The technical committee will promote exchanges among researchers from academia, industry and government. The purpose is to identify the technologies and technical approaches to advance and mature the field of aerial robotics. ([http://www.flyingrobots.org/](http://www.flyingrobots.org/))

✓ **UAV Swarm Project:** The Aerospace Controls Laboratory at MIT are investigating techniques to execute continuous missions (24-7) using multiple intelligent autonomous vehicles capable to charge automatically and many other functions useful for lots of applications. ([http://vertol.mit.edu/](http://vertol.mit.edu/))

✓ **Delft University of Technology:** The Faculty of Aerospace Engineering has developed during the last years studies to design and control Micro Air Vehicles. Its latest success was the The DelFly Micro, which only weighs 3 grams and has a size of 10 cm from wing tip to wing tip. This makes it the smallest flying ornithopter carrying a camera in the world. The DelFly Micro is very useful for the progress of science. Designing it already brought together knowledge from fields such as aerodynamics, mechanics, electronics, and camera technology. Furthermore, the DelFly Micro is an ideal platform for studying both the aerodynamics and autonomy of small, flying ornithopters. ([http://www.tudelft.nl/](http://www.tudelft.nl/))

The aerodynamic configuration for mini-UAV is rapidly changing. In the next figures we present a selection of recent aerial vehicles.

![Figure 1.10: Four-Rotor rotorcraft and X6 Tri-rotor from Draganflyer Innovations Inc (Credits [8]).](image)
The future tendency in MAV’s field, apart from focus on optimization of its structural design providing lighter vehicles with higher levels of autonomy, has a clear objective: the creation of swarm-bots.

They are expected to be Micro Aerial Vehicles with the capability of self-assembling, self-organizing, and metamorphosis between them and their systems. Such an approach finds its theoretical roots on recent studies in swarm intelligence, i.e., in studies of self-organizing and self-assembling capabilities shown by social animals.

These swarms of Micro Aerial Vehicles (SMAVs) are expected to be capable of multiple operations:

- Autonomously establishing emergency wireless networks between multiple ground-users in a disaster area (Figure 1.13). Such SMAVNETs could replace damaged, inexistent or congested networks and can play an
important role in disaster mitigation. The aerial nature of the system is interesting in that it allows for line-of-sight transmissions between MAVs, which is more energy-efficient than communication in cluttered environments at ground level. Furthermore, MAVs can fly over difficult terrain such as flooded areas or debris.

**Fig. 1.13** Emergency wireless network established by a swarm of MAVs [12]

- Search, monitoring, and path finding in built environments (Figure 1.14). Operating in swarm formation, as honeybees do, could efficiently explore built environments, locate predefined targets, and guide other robots or humans

**Fig. 1.14** Artistic example of a swarm of MAVs used in an airport [12]

Apart from these swarms of Micro Aerial Vehicles, other future expectative is the development of control strategies and neuromorphic chips for autonomous micro-flyers capable of navigating in confined or cluttered areas such as houses or small built environments using vision as main source of information. Flying in such environments implies a number of challenges that are not found in high-altitude, GPS-based, unmanned aerial vehicles (UAVs). These include small size and slow speed for maneuverability, light weight to stay airborne, low-consumption electronics, and smart sensing and control. It is expected that neuromorphic vision chips and bio-inspired control strategies are very promising methods to solve this challenge.
1.2 Objectives

The goal of our project was to study and develop the techniques to design and construct a Vertical Take-Off and Landing Micro Air Vehicle (VTOL MAV) capable of indoor and outdoor flight and maneuvering. Particularly VTOL MAVs present the opportunity to remain stopped during its flight to obtain some kind of data and information of the environment on a particular area in detail.

A number of requirements were set and assumptions made at the beginning of the project which drove development of the MAV in a particular direction:

- **Autonomous/controlled operations**: Capable to operate in two different modes, remote controlled like a simple RC aircraft or in autonomous operation, including take-off, hover, and landing capabilities, through pre-programmed flight paths. This implies that at least an autopilot must be considered into the design.

- **Bi-rotor or Tri-Rotor configuration**: Apart from being a natural choice for a requirement of low speed flight and hovering capability, this configuration must be an easy and feasible solution for indoor and outdoor applications. It will be potentially suitable for a large number of applications like inspections of bridges or for photography purposes. It was also possible to choose a quad-rotor configuration, but nowadays this solution is one of the most studied and we wanted to make research in new areas.

- **Electric power system**: Electric power cells are reliable, silent and clean, and thus suitable even for indoor operation; the energy and power density of the last generation of Lithium Polymer batteries allows for considerable endurance even for outdoor applications.

- **Maximization of payload and endurance**: Efficient experimentation of the design of the platform will be necessary to ensure robustness of the structure, long flight times, and the possibility to embed sensors and perceptive devices of acceptable quality.

- **Prototypes**: sub-optimal solutions can be initially accepted if rapidly available, while optimization of each component can be left to later design iterations.

The choice of configuration was primarily driven by the need to maximize payload, flight time and design efficiency (design difficulties vs. final performance). Two configurations were analyzed in the preliminary design phase: a hybrid vehicle with two vertical rotors to provide VTOL operations and an extra horizontal rotor to operate also in HTOL flights. This configuration was most efficient: depending on the operation environment the vehicle could use VTOL or HTOL maneuvers, maximizing the power endurance. At last, this configuration was discarded due to its structural complexity and doubtful performance (low payload weights).

So, we decided to study two cases: Bi-Rotor and Tri-Rotor configurations in order to study and compare its performances.
In the case of Bi-Rotor (Figure 1.15) the idea was very simple. If the two rotors were properly tuned (same speed and opposite rotation direction) and the weight was well balanced, the vehicle would take-off without any deviation on its operation axis. The vehicle could be provided of servo actuators that produce the capability to change the angle of the rotors plane independently to obtain yaw or pitch movements. Roll movement was considered not important for the vehicle performance, so we decided not to take it into account.

On the other hand, to make a stable Tri-Rotor (Figure 1.16) we assessed two options: a design based on two arms with the same length and the third longer in order to compensate the momentums, tuned at 120°; or three arms with the same length with a pair of motors on each one to produce counter-rotating momentums. At last we decided to choose the second option because it provided us the opportunity to make negligible the residual momentums and, tuning the motors in pairs we could get roll, pitch or yaw movements.

**Fig. 1.15** Our Bi-Rotor basic idea

Tuning the rotors properly we obtain equal momentums on each arm \( M \rightarrow -M \) and the predominant force is the total Lift as \( L_L = L_R \).

**Fig. 1.16** Our Tri-Rotor basic idea

Tuning each pair of rotors properly we obtain zero net momentum on each arm \( M \rightarrow -M \) and the predominant force is the total Lift. To maneuver we have to tune one or two pairs of rotors different than the others.
It was also important to think in the Reynolds number. It is a dimensionless number that gives a measure of the ratio of inertial forces ($V_p$) to viscous forces ($\mu/L$) and, consequently, it quantifies the relative important of these two types of forces for given flow conditions. It is an important factor to perform dimensional analysis of fluid dynamics. The fundamental low Reynolds number issue argued for as large a vehicle as possible to understand the aerodynamic processes that influence the vehicle. But, as our objective was to build something classifiable as “micro” or, at least, small enough, we adopted experimental techniques to understand the involved processes and extract some conclusions later.

In this project both of these designs will be developed in next chapters.

### 1.3 General MAV architecture

Figure 1.17 includes the main parts of a generic UAV. This does not mean all of them must follow this exact scheme, but these are the main parts. We will use this as a little generic introduction to the UAV design, and in Chapter 3 we will specify each of these parts for our concrete UAV.

![Fig. 1.17 General MAV diagram](image)

#### 1.3.1 Airframe

This includes the “physical” parts of the UAV, such the control surfaces to control the vehicle and all the actuators which will move them, such as the
servos and the motors. All of this is determined by the UAV structure, which will hold all the components.

The actuators receive the information given by the receptor or the microcontroller (flight controller) and actuate on the control surfaces.

1.3.2 Communication

The communication bloc includes all necessary things for establishing a link between the UAV and the base station (mobile or not). It can be used for controlling the vehicle, receiving data from the onboard sensors, etc… When developing the communication system you can design it from 0, defining your own protocol or buy a commercial solution such an RC transmitter/receiver pack. This has the advantage of possibly being quicker, but you also have limitations you wouldn’t have designing your communication system.

The transmitter sends the signal using its antenna and it is received by the receiver which will make it available to the microcontroller for being processed.

1.3.3 Processor

This is the brain of our vehicle; it will receive data from the onboard sensors and the receiver. It will be a microcontroller with preloaded software which will control the data flow and act on the different actuators based on these readings.

1.3.4 Sensors

This is the most configurable part of the entire system. Depending on the sensors we equip our UAV with; it will be able to perform different actions.

All the sensors transform a physical variable (distance, accelerations, pressure, temperature, etc.) into a change of intensity, capacity or voltage which is sent can to the microcontroller where the information is extracted.

1.3.5 Power

A full study of the system consume is needed in order to choose a correct battery with the adequate electric capacity. It is so important, because this value will determine the time our vehicle can be flying.

Also, a UAV needs different voltage values for its different parts, and in order to save weight it is better to have a power conditioning system, which will provide the needed values of tension, than having a different battery for each of these values (see Chapter 3.5 for details).

Following this architecture we will see our specific MAV parts on Chapter 3.
CHAPTER 2. DESIGN AND DYNAMICS

2.1 Structure

In this chapter we present the reason of the design choices taken for each vehicle (Bi-rotor and Tri-rotor).

Detailed mechanical explanations will be made in order to provide a source to make future reproductions, modifications or copies of the proposed designs.

2.1.1 Bi-Rotor Structure

In order to develop the structure design we set up some requirements. We needed something light, enough flexible to support low/medium crashes, easy to rebuild or repair and simple. Aerodynamically, design was not as a critical factor as weight because in VTOL MAVs lift is not provided by the fuselage or wings but for the engines and actuators. Also it had to have enough free space for its components (battery, receiver, servos, engines, sensors...), and for a future payload (wireless camera, environmental sensors...). With these requirements and thinks, we decided to adopt a design as shown in Figure 2.1.

![Bi-rotor scheme](image)

**Fig. 2.1 Bi-rotor scheme**

It accomplished all our previous requirements and had the advantage to be enough simple to be built by ourselves without critical complications allowing us to make future modifications in order to adapt new systems.

Four types of materials were required: plywood, plastic, carbon fiber and steel.

- **Plywood:** Our first think was to find a material to build the structure floors which allowed us to work and model it easily. The most important reason
for using plywood instead of plain wood was its resistance to cracking, shrinkage, twisting/warping, and its general high degree of strength.

- **Plastic:** Seeing the power of the engines and its high rate of RPM we noticed the necessity to create a propeller protector to avoid people from being hurt. We decided that this protector could be made of plastic to be enough resistant to support crashes having low weight. Also, after some test crashes we saw the necessity to modify a little bit the initial design and reinforce some of the floors with another type of material with similar degree of strength that plywood but higher degrees of tenacity and ductility. To satisfy this necessity we re-used a retail of plastic from a solar installation.

- **Carbon Fiber:** It is a material consisting of extremely thin fibers about 0.005–0.010 mm in diameter and composed mostly of carbon atoms. The carbon atoms are bonded together in microscopic crystals that are more or less aligned parallel to the long axis of the fiber. The crystal alignment makes the fiber very strong for its size. Carbon fiber has many different weave patterns and can be combined with a plastic resin and wound or molded to form composite materials such as carbon fiber reinforced plastic (also referenced as carbon fiber) to provide a high strength-to-weight ratio material (Figure 2.2). The density of carbon fiber is also considerably lower than the density of steel, making it ideal for applications requiring low weight. The properties of carbon fiber such as high tensile strength, low weight, and low thermal expansion make it very popular in aerospace and civil engineering.

- **Steel:** As we will see, some parts of the structure had critical requirements of higher strength-to-weight without suffering deformations. So we had to use steel threaded bars (Figure 2.3) to make it possible.

To design the vehicle we used Solid Edge® V18 from UGS, a Computer-Aided Design (CAD) program which provides solid modelling, assembly modelling and drafting functionality for engineers. This allows the user to create complex multiple pieces objects in a 3D environment. All the figures shown in this document has been made with this software.

First of all we created the arms where the motors had to be adapted. We used circles of plywood, carbon fiber bars of 2 mm and 5 mm and a retail of plastic to make the propeller protector (Figure 2.4).
The most important part of the arms was the joining between the arm and the servo. To build this piece we modify and fixed a cogwheel bought with the servos (Figure 2.5).

Second step was to create the body (Figure 2.6). This was the most important part of the vehicle so, after some design iterations, the decision of using steel threaded bars instead of carbon fiber ones was made. This also allowed us to use nuts to fix the planes instead of contact glue which made easily to repair or modify the design later. The upper level was made only of plywood because it is not exposed to big crashes and it was the best choice to cut and obtain the desired form properly. The medium level was exclusively of plastic because it has to have enough ductility to support the torsions of the steel bars without breaking. Finally, the inner level was composed by two pieces, one of plywood and one of plastic because we tried to obtain a mix of the properties of the two materials to have enough resistance to last the weight of the entirely structure without losing the form, but trying not losing a little bit of ductility, what was important because this part was the one which suffer more stress.
Some detailed areas have to be shown in this part. It is the case of the joint between the arms and the body (Figure 2.7). Physically, the critical point is the joint of the arm with the servo actuator, because all the Lift force generated by the motors is transmitted entirely to the servo and not to the structure. To avoid that we made a steel piece designed to create a previous contact point between the arm and the structure. By physical laws, now the force transmitted to the actuator is small enough to not affect its work.

Fig. 2.7 Force transmission point

Another important point to take into account was to prevent the arm to separate from the body and the servo actuator. To do that a small semi-cylinder was designed to delimitate the movement of the arm (Figure 2.8).

Fig. 2.8 Arm separating avoidance

Finally, assembling all this parts, and adding some landing protectors on each leg, the designed prototype was as shown in Figure 2.9 (see Annex 2 for plans) and the final, real Bi-Rotor is shown in Figure 2.10.
Fig. 2.9 Bi-Rotor design

Fig. 2.10 Bi-Rotor photo
The dimensions of the design are determined by the determining components such as motors, blades, battery, receptor and electronics.

As shown later we used 20.5 cm diameter blades, so the minimum weight was of 41 cm to avoid propeller collision. Even so, a safety margin has to be added and also we wanted to minimize the amount of blade that passes over the vehicle body because of aerodynamically problems it could generate. So finally, a 60 cm weight gave us the desired results.

On the vertical axis, we had more freedom when choosing its dimensions. The main cylindrical body height is designed in order to shelter battery and receiver on its bottom floor, so with the dimensions of its components we got the needed height of that floor. The upper floor was made for the electronics and here we estimated the needed space.

Finally, we added the legs, which dimensions have been chosen by checking the stability they gave to the structure, but any other length could be perfect for its purposes. These types of legs are simple enough and with three of them we proportionate a perfect plane on which the vehicle can stand.

### 2.1.2 Tri-Rotor Structure:

This second MAV was focused using the same main design parameters than our Bi-Rotor, low weight, resistant to medium crashes and easy to repair, rebuild or make future modifications. Nevertheless, this one had one parameter that had to be specially studied: structure perfect symmetry and stability.

Due to the fact that in this Tri-Rotor we did not assume the use of servo actuators, the compensation of strange movements originated by structural asymmetries, although possible with different turn rates of the engines, was much more complicated that in the Bi-Rotor. As we have seen, having the arms orientated with angles of 120° with the same length was a critical design necessity.

We proposed a simple structure design as shown in Figure 2.11:

![Tri-Rotor Diagram](image)
The contemplated materials to be used in this second design were a little bit simpler than before, focused only on carbon fiber and plastics.

This time we searched in the catalogues of the most important manufacturers of RC aerial vehicles to find existent parts and solutions that could help us to create our new structure. In this research we found two interesting ones that allowed us to conform the mainstays of the structure and the engines mount, sold by DraganFly Innovations INC (see Figure 2.12).

![Fig. 2.12 Draganflyer motor mount and vertical riser leg](image)

Then, our design was to be composed by: three carbon fiber sticks with 4 mm of diameter, three motor mounts and vertical riser legs, two hexagonal pieces of plastic and some brass pieces to join them (Figure 2.13).

![Fig. 2.13 Tri-Rotor CAD parts](image)

The first step was to obtain rigid non-moveable arms with 120° between them. To do that a brass piece was created, consisting in three cylinders with an inner diameter of 4 mm to make possible the union with the carbon fiber sticks, mounted at 120° each (Figure 2.14). With this piece and the vertical riser legs, we assured a total arms stability and inflexibility.
Next stage consisted in the fitting of the upper plastic cover which closed the structure. To make it possible a connection brass pieces were created (Figure 2.15). Note that it is necessary the use of M2 screws in order to fix them all.

Once we had that, the structure was finished, obtaining a result as shown in Figure 2.16 compared to the real Tri-Rotor (Figure 2.17).
2.2 Dynamics

Unless we have used experimental techniques to develop the vehicles, it is also important to determine some basic motion equations to know the physical processes involved in its maneuvering.

VTOL MAVs are among the most complex flying objects because their flight dynamics is inherently nonlinear and they have strong couplings of all the variables. As we have seen, they have however the ability to hover which is required in some applications. Small-scale unmanned vehicles may be expected to display considerably different dynamic response than a full-scale one. The dynamic model has been modeled, in general, by Newton equations of
motion. We can find in the literature complete dynamic models; all of these contain the complete behavior of the some kinds of these vehicles in different flight conditions (hover, vertical and forward flight). However, the main problem of these dynamic models is the difficulty in designing a “simple” control algorithm due to the complexity and the existence of cross-terms in the model equations. To solve this problem, some researchers have proposed simple dynamic models for the controller design of these aircraft. The first attempt to control small VTOL vehicles was done by using linear techniques. Nevertheless, the nonlinear nature of them has to be taken into account in the controller design if one wants to improve their performance. Backstepping techniques were used for the design of a nonlinear control law. Recently, Isidori et al. [14] presented a nonlinear robust regulation in order to control the vertical motion of a simple VTOL vehicle like a standard helicopter.

In recent times, NASA has started the research on flying-qualities such as directional stability, lateral oscillations turn characteristics and speed stability for a tandem vehicle with nonoverlapped-rotors, with the purpose of reducing the disadvantages of this type of instability. These experiments have been essential for the development of the VTOL aircraft.

In this section we will see a simplified vision of the processes evolved on our vehicles maneuvers’, simplified models based on Newton equations of motion [11] and a little bit of introduction to control techniques.

2.2.1 Bi-Rotor

2.2.1.1 Maneuvers diagrams

We have the system shown in Figure 2.18. Where $F_1$ and $F_2$ are the lift forces produced by the engines, $\beta$ is the inclination angle of craft arms (caused by the servos); $\Omega$ is the residual elevation angle produced by arms flexion, $W$ is the craft weight focused on the center of gravity, $D$ the drag force and $M_i$ and $M_i'$ the momentums originated by the motors.

![Basic Force diagram](image)
The use of counter-rotating blades allows us to ideally cancel the motor momentums if both are tuned properly. However, asymmetries and differences between engines’ performances may cause a momentum that would have to be cancelled by de control surfaces.

Once said that, we will study the vehicle’s movement divided into two parts: translation and rotation.

### 2.2.2 Translation

The study here is made by dividing the movement into the vertical and the horizontal plane.

**Vertical maneuvers:**

If we study only the vertical axis, we can reduce the system to the scheme in Figure 2.19:

As we observe, both inclination and flexion contribute to reduce the lift. So, it is important to take them into account to obtain accurate results. Note that each effect takes influence on both arms of the craft.

We will start using Newton’s second law, which says that the net force on a particle of constant mass is proportional to the time rate of change of its linear momentum.

\[
\sum \vec{F} = \frac{d}{dt} (m\vec{v}), \quad \text{which can be written as:} \quad \sum \vec{F} = m\ddot{\vec{a}}
\]  

(2.1)

As we are studying the vertical movement, the drag force and the weight of the vehicle vector’s are completely parallel, which allows us to simply add or subtract its modules. The same way, we can do it with the vertical projection of \( \vec{F}_\beta \). So, considering only effects introduced by \( \beta \), and reducing all the craft mass to the center of mass, we have that:
\[ |\vec{F}_{\beta 1} \cos \beta + |\vec{F}_{\beta 2} \cos \beta - |\vec{W} - |D = m|\vec{a}_y| \]  \hspace{1cm} (2.2)

As \( \beta \) module value will be always equal due to design parameters, the projection on the Y axis will be equal in both arms, so we can add the two forces:

\[ |\vec{F}_{\beta 1} \cos \beta = |\vec{F}_{\beta 2} \cos \beta \rightarrow 2|\vec{F}_{\beta} - |\vec{W} - |D = m|\vec{a}_y| \]  \hspace{1cm} (2.3)

Finally, if we refer this expression with the Y component of \( \vec{F}_{\beta} \), which is: \( F_{\beta y} = |\vec{F}_{\beta} \cos \beta \) and isolating the acceleration:

\[ |\vec{a}(\beta, \%_{e.power})_y| = \frac{2|\vec{F}_{\beta}(\%_{e.power})| \cos \beta - |\vec{W} - |D}{m} \]  \hspace{1cm} (2.4)

The previous expression describes the behavior of the system on Y-axis quite well, but we could also add the flexion effects to be more accurate. These effects are caused mainly by the weight of the engines and the arm structure applied in a distance \( d \) from the fixed point.

To obtain the net force after that, we will proceed similar than before. By decomposition of forces we have that:

\[ F_{\Omega y} = F_{\Omega} \cos \Omega \]  \hspace{1cm} (2.5)

So now, the acceleration of the craft will be:

\[ |\vec{a}(\beta, \%_{e.power})_y| = \frac{2|\vec{F}_{\beta}(\%_{e.power})| \cos \beta \cos \Omega - |\vec{W} - |D}{m} \]  \hspace{1cm} (2.6)

This can be also expressed as:

\[ |\vec{a}(\beta, \%_{e.power})_y| = \frac{2|\vec{F}_{\beta}(\%_{e.power})| \cos \beta \cos \Omega - |D|}{m} - g \]  \hspace{1cm} (2.7)

Being \( F \) the lift force depending on the angles \( \beta \) and \( \Omega \) and on the engine power percentage; \( g \) the gravity acceleration (9.81 \( m/s^2 \)), \( D \) the drag force and \( m \) the entire mass of the MAV.
In our particular case, the design has been focused on make torsion effects negligible. So we can consider $\Omega = 0$, which makes Equation 2.4 a perfect approximation.

We need to know $|\vec{F}_\beta (\%_{e\text{power}})|$ and $|\vec{D}|$ values in order to use these formulas. $|\vec{F}_\beta (\%_{e\text{power}})|$ is given in Graph 2.1 and can be approximated with a $R^2$ correlation parameter of 0.991 by the following second order equation:

$$|\vec{F}_\beta (\%_{e\text{power}})| = -4.1x^2 + 7.64x + 0.107 \ [N]$$

(2.8)
What this formula computes is the difference between the real vertical lift force without motors deflection and the vertical component of the force generated by the engines once a deflection has been made. With that we can see how much lift force we will loss for a determined angle of deflection.

As we said before, an acceptable approximation can be made by neglecting the Ω flexion effects.

Making a graph with this expression we get Graph 2.2.

![Graph 2.2 Lost Lift](image)

**Graph 2.2** Lost Lift

Until now, we have seen the effects of the resultant forces, but we know that all perpendicular force applied at a distance $d$ from the center of mass originates a momentum (Figure 2.20):

$$M = |\vec{F}| \cdot d \cdot sen(\alpha)$$

(2.11)

![Fig. 2.20 Momentums](image)
This has no effect in our system due to each arm originate a momentum with the same module but opposite signs, so ideally they are cancelled but possible asymmetries can generate residual momentums.

With all these calculi we can verify that, ideally, the designed vehicle will be able to fly just for Newton laws. Also we found the tool to compute the maximum vertical speed at which our MAV could ascend, the mathematical expression of the engines power depending on its \% power and the loss of lift that produces the inclination of the motors.

**Horizontal maneuvers:**

Due to Bi-Rotor design, the movement through the horizontal plane is obtained by deflecting the motors in the same direction. Note that different angle deflection could be achieved for each motor and the craft would advance and rotate at the same time, but for simplicity we take the same angle for both motors, as shown in Figure 2.21. In this case, each engine originates a horizontal component of force ($F_{\beta x}$) in the same direction that will make the vehicle move.

\[
2 |F_{\beta x}| \sin \beta - |D| = m |\ddot{a}|
\]  
(2.12)

Where the drag term could be found experimentally as we said on page 25.

\[
|\ddot{a}(\beta, \%_{e,\text{power}})|_z = \frac{2 |F_{\beta x}(\%_{e,\text{power}})| \sin \beta - |D|}{m}
\]  
(2.13)
2.2.3 Rotation

We have seen how to move up and down throw the Y-axis (vertical plane) and forward and backward in the Z-axis (horizontal plane), but now it is time to see how to perform the last important maneuver: rotate in the horizontal plane (X/Z-axis) (Figure 2.22).

To do that, we will apply the same amount of inclination to the rotors but in different directions. This will cause an angular momentum that will make the craft to roll over the horizontal plane. Note that now, and in our ideal situation (without consider any perturbation), the projected forces shown in the previous paragraph are cancelled between them.

Fig. 2.22 $\beta \neq 0$ with non-symmetrical arms movement

So, our vehicle is going to describe an accelerated circular movement over the Y-axis (roll) (Figure 2.23).

We can see the system as:

Each force is associated with a tangential acceleration that can be used to find the angular velocity $\omega$ of the rotation.

Fig. 2.23 Accelerated circular movement

Knowing that $2|\vec{F}_{\beta x}| - |\vec{D}| = m\ddot{a}_t$, we have that:

$$\left|\ddot{a}_t(\beta, \%\text{e, power})_{roll}\right| = \frac{2|\vec{F}_{\beta}(\%\text{e, power})|sin\beta - |\vec{D}|}{m} \quad (2.14)$$
Which relating with angular acceleration $\alpha$ is:

$$\left| \hat{\alpha}(\beta, \%e_{power})_{roll} \right| = \frac{2 |\vec{F}_\beta(\%e_{power})| \sin \beta - |\vec{D}|}{m \cdot R}$$  \hspace{1cm} (2.15)

This formula works with the assumption of a punctual mass, but in fact it is not what we have. We neglect the body mass because it is on the spinning axis and we consider it a punctual mass, so only the arms masses are taken into account.

### 2.2.3.1 Newton-Euler Model [11]:

Our Bi-Rotor MAV, as we have seen, uses two counter-rotating rotors of equal size and loading, so there is no net yaw moment on the helicopter because the torques of the rotors are equal and opposing.

Pitch and yaw movements are achieved by differential change of the main rotors thrust magnitude produced by servo’s inclination. And roll movement could be obtained by applying different levels of thrust on both rotors and compensating the residual moments with servo’s action (this is going to be neglected in this first prototype).

For simplicity we will present here the dynamic model in hovering. It is important to consider that the operation of two or more rotors in close proximity will modify the flow field at each, and hence the performance of the rotor system will not be the same as for the isolated rotors. We will not consider this phenomenon to simplify the dynamical model (Figure 2.24).

In order to obtain the final dynamic equations we have separated the aerodynamic forces into two groups. The first group is composed of translational forces and the second is related to the rotational forces of motion.

![Model diagram](image)

**Fig. 2.24** Model diagram
**Translational Forces:**

Denote by $T_L$ and $T_R$ the thrust generated by the right (R) and left (L) rotors respectively. These forces, as we have seen, have no $E_1$ component, so the thrust vectors are defined by:

$$T_R = |T_R|^2 E_2 - |T_R|^2 E_3, \quad T_L = |T_L|^2 E_2 - |T_L|^2 E_3 \quad (2.16)$$

By simple geometric analysis we obtain expressions in terms of $\beta$, where $\beta$ is the angle between the axis $E_3$ and the actual thrust vector:

$$T_R = |T_R| \sin \beta_R E_2 - |T_R| \cos \beta_R E_3 \quad (2.17)$$
$$T_L = |T_L| \sin \beta_L E_2 - |T_L| \cos \beta_L E_3 \quad (2.18)$$

The thrust vector can be represented by the expression:

$$T_i = |T_i| \begin{bmatrix} 0 \\ \sin \beta_i \\ -\cos \beta_i \end{bmatrix} = |T_i| \begin{bmatrix} 0 \\ \beta_i \\ -1 \end{bmatrix} \quad (2.19)$$

Where $i = R$ or $L$, and considering sufficiently small values of $\beta_i$. Another force applied to the MAV is the gravitational force given by $f_g = m g E_z$, where $m$ is the complete mass of the vehicle and $g$ is the gravitational constant. The above expression is defined in the inertial frame $I$. In terms of the body fixed frame, it is necessary to multiply $f_g$ by the inverse of the rotation matrix $R$ that represents the orientation of the body-fixed frame $C$ with respect to $I$.

So, denoted by $f$, the total translational force applied to the MAV expressed in the inertial frame $I$ is:

$$f = (|T_R| \beta_R + |T_L| \beta_L) R E_2 - (|T_R| + |T_L|) R E_3 + m g E_z \quad (2.20)$$

**Rotational Forces: torques and anti-torques**

The torques generated by the thrust vectors $T_R$ and $T_L$ are due to separation between the centre of mass CG and the rotor hubs (called $\tau_R$ and $\tau_L$ respectively). The gravitational force does not generate a torque since the helicopter is free to rotate around its centre of mass. Before beginning, it is necessary to define the measured distance from the centre of mass of the vehicle to the hubs of the two rotors (denoted $l_R$ for the right rotor and $l_L$ for the left rotor). If we express these vectors in terms of the body fixed frame we have,
\[ l_R = l_R^1 E_1 - l_R^3 E_3 \]  
\[ l_L = -l_L^1 E_1 - l_L^3 E_3 \]  

(2.21)  

(2.22)

The torques applied to the airframe by the thrust vectors are defined by,

\[ \tau_R = l_R \times T_R \]  
\[ \tau_L = l_L \times T_L \]  

(2.23)  

(2.24)

The total torque generated by the nose and tailing rotors is given by,

\[ \tau_{RL} = \tau_R + \tau_L = \left[ \begin{array}{c} l_R^2 |T_R| \beta_R + l_R^2 |T_L| \beta_L \\ l_R^1 |T_R| - l_R^1 |T_L| \\ l_R^3 |T_R| \beta_R - l_L^3 |T_L| \beta_L \end{array} \right] \]  

(2.25)

Additionally, the aerodynamic drags on the rotors generate some pure torques acting through the rotor hubs. Assuming same levels of thrust for each rotor, the anti-torques are defined by,

\[ Q_R = |Q_R| E_3, \quad Q_L = -|Q_L| E_3 \]  

(2.26)

Finally, the total torque applied (expressed in the body fixed frame) is given by,

\[ \tau = \tau_{RL} + |Q_R| E_3 - |Q_L| E_3 \]  

(2.27)

Finally, substituting the total force and torque in a Newton basic model we have,

\[ \dot{x} = v \]  

\[ m \dot{v} = (|T_R| \beta_R + |T_L| \beta_L) R E_2 - (|T_R| + |T_L|) R E_3 + mg E_z \]  

\[ \dot{R} = R \Omega \]  

\[ I \dot{\Omega} = -\Omega \times I \Omega + |Q_R| E_3 - |Q_L| E_3 + \tau_{RL} \]  

(2.28)
2.2.4 Tri-Rotor

In this vehicle, each rotor axis is actually formed by two rotors, as shown in Figure 2.25.

Blades are disposed in order to make the lift force upwards, and because they are aligned can be added to obtain a global lift force for each rotor axis, which is for each pair of motors. Also they are counter-rotating, so both give an upward force while one is rotating clockwise and the other one anti-clockwise. This enables us to ideally cancel the momentums generated by the motors at the same speed, by supposing that its momentum values are exactly equal and with opposite sign.

So with the simplifications in equation (2.29) the previous picture is simplified as shown in Figure 2.26.

\[ F_i = F_{i1} + F_{i2} \]
\[ M_i = -M_i' \]  

(2.29)

---

Fig. 2.25 Tri-Rotor arm

Fig. 2.26 Lift force produced by both motors
So, the general vehicle scheme dynamics is shown in Figure 2.27.

![Fig. 2.27 Tri-Rotor dynamics](image)

Let’s now look at the different maneuvers the vehicle is assumed to perform, and the dynamics associated to them by analyzing the translation and the rotation of the vehicle.

### 2.2.4.1 Translation

In this chapter we will look at the displacement of the vehicle through a 3D space, which can be simplified to a two planes movement (Figure 2.28).

![Fig. 2.28 Tri-Rotor translation movement](image)
The general forces sum is the following, but to analyze the dynamics of the translation maneuvers we will study the vertical and horizontal planes separately for simplicity.

\[ \vec{F}_1 + \vec{F}_2 + \vec{F}_3 - \vec{D} - \vec{W} = m\vec{a} \quad (2.30) \]

**Horizontal plane**

Due to the design of the Tri-Rotor, moving throw the horizontal plane must be done by changing the vehicle’s attitude because it have no moving parts that allow it to deflect the motors. So we need to give an angle to the lift forces with the vertical axis by incrementing the engine power in whichever of them we want, in order to generate a horizontal component which will be the responsible of making the craft advance (Figure 2.31).

So, the forces sum is the following:

\[ F_{1x} + F_{2x} + F_{3x} - D_x = ma_x \quad (2.31) \]

It can be expressed with the total lift force given by each pair of engines, which is the data we have.

\[ |\vec{F}_1| \sin \beta + |\vec{F}_2| \sin \beta + |\vec{F}_3| \sin \beta - D_x = m|\vec{a}_x| \quad (2.32) \]

Also due to design the angle of the forces are the same for each rotor, so the vehicle’s horizontal acceleration is,

\[ |\vec{a}_x| = \frac{(|\vec{F}_1| + |\vec{F}_2| + |\vec{F}_3|) \cdot \sin \beta - D_x}{m} \quad (2.33) \]

where for \( \beta \neq 0 \rightarrow |\vec{a}_x| > 0 \), which is the desired behavior.

**Vertical plane**

This is the plane which makes a vehicle a flying vehicle. From Figure 2.31 the following forces sum can be extracted:

\[ F_{1y} + F_{2y} + F_{3y} - W - D_y = ma_y \quad (2.34) \]
Proceeding as we did on horizontal axis, we get:

\[
|\vec{F}_1| \cos \beta + |\vec{F}_2| \cos \beta + |\vec{F}_3| \cos \beta - W - D_y = m|\vec{a}_y|
\]  

\[
|\vec{a}_y| = \frac{(|\vec{F}_1| + |\vec{F}_2| + |\vec{F}_3|) \cdot \cos \beta - W - D_y}{m}
\]

With this formula we can compute the vertical acceleration of the vehicle, or the total force needed for reaching a determined acceleration. Also we see that the maximum total force will be reached with \( \beta = 0 \).

### 2.2.4.2 Rotation

We said before that momentums were annihilated due to the counter-rotating blades. Until now, this is perfect for our purposes, but in order to rotate the vehicle on its center axis we need to change that. By increasing or decreasing all the momentums in the same direction we can achieve a rotation of the vehicle, which can be useful for vision applications, where the user may want to “look around” without moving its position (Figure 2.29).

![Tri-Rotor rotation](image)

**Fig. 2.29 Tri-Rotor rotation**

For example, if we want to turn a little bit clockwise, we will give the upper motors more power so,

\[
M_1 + M_2 + M_3 > M_1' + M_2' + M_3'
\]

\[
M_{total} = (M_1 + M_2 + M_3) - (M_1' + M_2' + M_3')
\]

So, our vehicle is going to describe an accelerated circular movement over the Y-axis (roll).
Once analyzed the basic dynamics of our vehicles' maneuvers, it is demonstrated that we can move them under control in a whole 3D environment, which is one of the main requirements of an MAV.

## 2.3 Basics of control

Although it is not the purpose of this project, a plant for the MAVs can be computed with all previous equations in order to develop a control system as explained as follows.

To make a simple approach to control aspects [13] it is necessary to talk about the model plants. This term is a mathematical representation of the relation between the input and output of a system. With that we can model the system behavior produced by some excitation (ex. a gust of wind). This relation is given by the following expression (usually in Laplace domain):

\[
G(s) = \frac{Y(s)}{R(s)}
\]  

(2.38)

Being \( G(s) \) the transfer function of the system, \( Y(s) \) the output signal and \( R(s) \) the input signal (Figure 2.30).

![Block diagram](image)

**Fig. 2.30 Block diagram**

The last figure is known as open loop transfer function because it has no feedback of the output signal to correct the possible errors. So, what our system really needs is a closed loop strategy (Figure 2.31).

![Closed loop system](image)

**Fig. 2.31 Closed loop system**

Where \( \varepsilon(s) \) is the difference between the entry and the exit signal, the error.

In this case, the closed loop sends the output signal to the beginning to analyze the differences with the input and reduce the error value.

Nevertheless, to control this feedback is common to use a PID controller to correct the error between the average and the desired value. This controller in
fact is a combination of three sub-controllers: Proportional, Integral and Derivative ones; and its function is given by the following expression:

\[ G_c = K_P + \frac{K_i}{s} + K_D s \]  \hspace{1cm} (2.39)

Where \( G_c \) is the PID function, \( K_P \) the proportional term, \( K_i \) the integral term and \( K_D \) the derivative term.

Next table shows the behavior of the system adding the controllers.

**Table 2.1 PID effect**

<table>
<thead>
<tr>
<th>Controller</th>
<th>Peak time</th>
<th>Overshoot</th>
<th>Settling time</th>
<th>Steady-state error</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Decrease</td>
<td>Increase</td>
<td>Small change</td>
<td>Decrease</td>
</tr>
<tr>
<td>I</td>
<td>Decrease</td>
<td>Increase</td>
<td>Increase</td>
<td>Eliminate</td>
</tr>
<tr>
<td>D</td>
<td>Indefinite (small decrease or increase)</td>
<td>Decrease</td>
<td>Indefinite (small decrease or increase)</td>
<td></td>
</tr>
</tbody>
</table>

Peak time \((t_p)\): Time necessary for the response to get the first impulse.

Overshoot \((M)\): Maximum peak amplitude of the system response.

Settling time \((t_s)\): Required time that a signal takes to establish its values in a determined error zone.

Steady-state error \((\varepsilon_s)\): Difference between the input and output signals.

Next figure shows these parameters. Note that they are expressed in the time domain.
In our particular case, it is possible to demonstrate that the plant of our vehicles will be given by a function with the following form in the Laplace domain [13],

\[ H = \frac{y}{A s^2} \]  

(2.40)

Being \( H \) the studied plant of the vehicle and \( y \) and \( A \) parameters depending on the concrete vehicle.

So, our system would describe a loop as shown in Figure 2.33.

Fig. 2.33 Bi-Rotor and Tri-Rotor system closed loop

Analyzing our plant and the PID controller equation (2.39) we can conclude that in our particular case the Integral part of the controller will not be needed due to the integral term is contained into the plant equation (Figure 2.34). So, what we would need to develop a control system for our two vehicles is just a Proportional-Derivative controller (PD).
With all this information we only expect to give a brief introduction to control techniques. To see more information and studies about this topic see [13] and [15].

\[ H = \frac{\gamma}{A S^2} \quad G_c = K_P + K_I + K_D S \]

**Fig. 2.34** Not needed Integral Controller
CHAPTER 3. HARDWARE

3.1 Actuators

The actuators are the onboard components that actually can perform some actions, such as rotational or translational movements. They are the ones which will allow us to modify the vehicle’s structure while flying, either using a remote control or an autonomous one.

In our MAV we have two different actuators (the most common ones): the motors, and the servo-motors. The main difference between them is on its operation. A electrical motor, as we all know, is designed to operate at a high rotational speeds describing a continuous movement going from its initial position \((0^0)\) to the same one \((360^0)\) while a servo can provide different angular positions with variable angle resolution and ranges. We will talk more about these two actuators on this chapter.

3.1.1 Motors

They are the critical part of the design, because based on the performances achieved by the chosen motor we will be able to lift more or less weight and different structure characteristics will be needed to hold it. Because of this we have done a study of different motors with different blades in order to choose the adequate model for our purpose.

After showing and talking about the obtained results, let us talk about two different types of electrical motors: brushed motors and brushless motors. In the last years, brushless motors are becoming the most used solution either on aeromodelling world or in other industrial applications. Let’s see why.

3.1.1.1 Brushed Motors

In a conventional (brushed) DC motor (Figure 3.1), the brushes make mechanical contact with a set of electrical contacts on the rotor (called the commutator), forming an electrical circuit between the DC electrical source and the armature coil-windings. As the armature rotates on axis, the stationary brushes come into contact with different sections of the rotating commutator. The commutator and brush system form a set of electrical switches, each firing in sequence, such that electrical power always flows through the armature coil closest to the stationary stator.

Fig. 3.1 Brushed motor
This has some problems or **disadvantages** most of them due to the mechanical friction generated between the brushes and the commutator:

- At higher speeds, brushes have increasing difficulty in maintaining contact. Large brushes are desired for a larger brush contact area to maximize motor output, but small brushes are desired for low mass to maximize the speed at which the motor can run without the brushes excessively bouncing and sparking.
- The brushes wear out.
- The brushes and commuter must be cleaned periodically.
- Friction from the brushes lead to shorter flight times and battery life, cause lower power to weight ratio and slows down the motor.

### 3.1.1.2 Brushless Motors

Some of the problems of the brushed DC motor are eliminated in the brushless design (Figure 3.2). In this motor, the mechanical “rotating switch” is replaced by an external electronic switch synchronized to the rotor’s position. So, the Electronic Speed Controller (ESC) takes care of switching the voltage of the electromagnets. Needless to say, the higher the switching frequency is, the faster the motor will spin.

The switching frequency is controlled by a Pulse-Width Modulation (PWM) signal (Figure 3.3).

What we see in Figure 3.3 is a PWM, where the Duty Cycle ($D_c$) is the duration of the pulse and $T$ de period of the signal. Most of the ESCs work with a 50 Hz frequency, which is equal to a 20 ms period. And $D_c$ has a range of useful values that will correspond to the motor’s maximum and minimum speed (which usually goes from 1 ms to 2 ms pulse duration).
The ESC receives the PWM (Figure 3.4), determines the duration of the pulse, and based on this value gives a determined switching frequency. As we will see deeper on the servo’s chapter, a 1 ms pulse is the minimum speed and 2 ms the maximum speed, where the middle values can be extrapolated linearly.

![Fig. 3.4 Electronic Speed Controller](image)

Each motor has its own performances, which mean that for a determined pulse width, each one will achieve a different RPM values. And due to that the conversion \( D_c - f_{\text{switching}} \) is done by the ESC, even with the same motor and changing the ESC the RPMs may be different. Because of this is, is pointless to give a generalist relationship between this values. We will give them later, with our chosen motor – ESC – blade set.

Brushless motors can be classified based in two aspects. The first we are going to see is if it have or not a hall sensor:

- **With hall sensor:** The motor is equipped with a hall sensor that measures the magnetic field inside the solenoid and determines the position of the rotor, which is given to the ESC. With this, a smoother behaviour of the motor is achieved on first rotations.

- **Without hall sensor:** The motor does not “know” the position of the rotor, it just changes the current from one phase to another with a frequency determined for the control signal the ESC receives. In this case some transitory is perceived when the motor starts rotating.

Without hall sensor type is the most used because it is cheaper and lighter.

The second classification type is based on which part of the motor rotates:

- **Inrunner:** Inrunner motors turn very fast and are much more efficient than outrunner motors. Inrunner brushless RC motors require a reducing gearbox between the motor and propeller, for this reason, the output speed and torque of the propeller can easily be "tweaked" to facilitate different flying characteristics by using different size gears.

The downside is added parts that can and do fail. The gears get stripped, and the gearbox shafts are easily bent.
✓ **Outrunner:** Outrunner RC motors spin much slower and provide much more torque than inrunner motors. The greatest benefit of an outrunner motor is the fact that a gearbox is not needed and this makes the airplane literally silent. Outrunner motors are much easier to mount.

The downside of an outrunner motor is that you have a very small window of propellers you can use. Outrunner brushless RC motors are also less efficient than inrunner motors. However, this shouldn't be a huge factor when determining the best motor for your RC airplane.

**Table 3.1. Brushed and brushless motors comparison**

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ADVANTAGES</th>
<th>DISADVANTAGES</th>
<th>TYPICAL APLICATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brushed DC</td>
<td>- Low initial cost</td>
<td>- High maintenance (brushes)</td>
<td>- Treadmill exercisers</td>
</tr>
<tr>
<td></td>
<td>- Simple speed control</td>
<td>- Low lifespan</td>
<td>- Automotive starters</td>
</tr>
<tr>
<td>Brushless DC</td>
<td>- Long lifespan</td>
<td>- High initial cost</td>
<td>- Hard drives</td>
</tr>
<tr>
<td></td>
<td>- Low maintenance</td>
<td>- Requires a controller</td>
<td>- CD/DVD players</td>
</tr>
<tr>
<td></td>
<td>- High efficiency (85 %)</td>
<td></td>
<td>- Electric vehicles</td>
</tr>
</tbody>
</table>

### 3.1.2 Servos

A servo (or servomotor) is an actuator device which has the ability to stay on any of its possible positions (which normally is 180°). It is composed by an electric DC motor, a reduction gear and a control circuit.

The main component of a servo is the electric DC motor, which turns at high speed but with low par, when applying a voltage between its terminals. In order to increase the torque of the device a reduction gear is added, transforming much of the angular velocity into torsion.

#### 3.1.2.1 Position control

The device uses a control circuit in order to position the motor at one point. The angular position is determined through a PWM. The reference point is set using a square control signal (PWM), the pulse width will determine this angle: a widest pulse will set the motor on a bigger angle and viceversa. What we see on Figure 3.5 is the control system, where dotted line represents a mechanical coupling and the continuous line represents electrical connections.
Initially, the error amplifier calculates the error position value, which is the difference between the reference and the actual position of the motor. A bigger error position value means this difference is bigger, so the motor will have to turn faster to reach the desired position and viceversa. If the servo is on the desired position, the error will be 0 and there will be no movement.

The error amplifier needs to subtract two analogical voltage values. The PWM control signal becomes then into an analogical voltage value, by using a pulse width – voltage converter. The other voltage value is the one related to the motor position, which is obtained using a potentiometer mechanically coupled to the motor’s gear box, so when the motor rotates, the potentiometer also will, changing the voltage introduced into the error amplifier.

Once the position error has been obtained, it is amplified and applied to the motor terminals.

3.1.2.2 Controlling the servo

The servo control consists on indicate its position through a PWM, where it depends on the duration of the high level of the signal. Each servo, depending on the trademark and model, has its own ranges of operation. For most of the aeromodelling servos, the pulse can vary between 1 ms and 2 ms, which position the motor in its extremes ($0^\circ$ and $180^\circ$, respectively).

The relationship between the pulse width and the position is completely lineal: 1.5 ms pulse duration is the central position and other duration values position the motor proportionally (Figure 3.6).
It is easy to see that for the case of the figure the duration of the pulse for reaching the desired angular position $\theta$ is given by the following formula:

$$ t = 1 + \frac{\theta}{180^0} \quad t = pulse\,\,duration[ms] \quad \theta = angular\,\,position[deg] \quad (3.1) $$

However, the angle value is limited to the range operation due to the physical limitation that the potentiometer imposes. The duration value, on the other hand, is not so restrictive, because if we insert a pulse longer than 2 ms, the servo will stay at its maximum position and the same for the other extreme.

For blocking the servo on a position it is necessary to continuously send him the signal with the desired position. This way, the control system will be working and the servo will stay quiet and will resist external forces that try to move it. If the pulses are not sent, the servo will stay “free” and any external force may move it from its position.

### 3.2 Motors & Blades Study

As we said before, we have to choose wisely the motors we are going to use, so for a wise decision we need information, and we have to get it. Here we will see the data we obtained and how we obtained it.

First of all we need to name the motors (Figure 3.7) and the blades (Figure 3.8):
## MOTORS

<table>
<thead>
<tr>
<th>Motor name</th>
<th>Type</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tail</td>
<td>Brushed</td>
<td>6.8 g</td>
</tr>
<tr>
<td>Quad_rotor</td>
<td>Brushed</td>
<td>47.7 g</td>
</tr>
<tr>
<td>DVD</td>
<td>Brushed</td>
<td>21.8 g</td>
</tr>
<tr>
<td>Scalextric</td>
<td>Brushed</td>
<td>27 g</td>
</tr>
<tr>
<td>Hacker</td>
<td>Brushless</td>
<td>32.9 g</td>
</tr>
<tr>
<td>Emax 5”-6”</td>
<td>Brushless</td>
<td>31 g</td>
</tr>
<tr>
<td>Emax 9”-10”</td>
<td>Brushless</td>
<td>54 g</td>
</tr>
<tr>
<td>Himax</td>
<td>Brushless</td>
<td>57 g</td>
</tr>
</tbody>
</table>

**Fig. 3.7** Tested motors

## BLADES

<table>
<thead>
<tr>
<th>Blade name</th>
<th>Weight</th>
<th>Diameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>3.0 g</td>
<td>10.38 cm</td>
</tr>
<tr>
<td>B1</td>
<td>0.9 g</td>
<td>8.21 cm</td>
</tr>
<tr>
<td>O1</td>
<td>0.4 g</td>
<td>6.5 cm</td>
</tr>
<tr>
<td>O2</td>
<td>1.1 g</td>
<td>10.19 cm</td>
</tr>
<tr>
<td>B2</td>
<td>9 g</td>
<td>31.3 cm</td>
</tr>
<tr>
<td>APC 8x3.8</td>
<td>7.3 g</td>
<td>20.5 cm</td>
</tr>
</tbody>
</table>

**Fig. 3.8** Tested blades
What we want to know is the weight that a determined couple of motor – blade can lift in function of some controllable parameter such as the RPM, the % power of the motor, or any other. Also we will look at the consumption for each one of these steps in order to calculate later which battery to use.

To get this data we have used a precision scale\(^1\) to get the lift weight. We have done it with the blade making the net force facing the ground in order to be able to read the weight directly from the scale display. Also we have done it this way because in normal position the scale does not give a correct value because the airflow exerts force on it. Let’s see both schemes on the following figures.

If we decide to do the lift force upwards, so the motor tries to climb (Fig. 3.9 left), the airflow that the blade is “moving down” hits the scales plate and it does not sense any change at all because the lift weight is annulled by the pressure that the airflow is doing. It may sense some change because these two values do not have to be equal, but it is not of our interest.

Instead of doing it this way, we can do it facing the lift to the ground (Fig. 3.9 right). Now, the airflow does not affect the scale readings because the air mass is moved to the free space and what we are reading on the display is the lift force the blade is actually giving. Needless to say, that is how we tried our motors’ performances.

So let’s see at the obtained results. We first have studied the brushed motors with different blades (Figure 3.10) and then we have passed to the brushless motors using the blade “APC 8x3.8” because it is no sense using the smaller ones with these motors (Figure 3.11).

---

\(^1\) The used scale has been “COBOS precision h D-4000-SX” with a maximum weight of 4 Kg, 0.1 g sensibility, a working temperature 0 ° and 40 °.
Obtained data from tested brushed motors:

**Quad_rotor motor**

**DVD motor**

**Scalextric motor**

**Tail motor**

Fig. 3.10 Tested brushed motors
Obtained data from tested brushless motors:

**Emax 9" - 10" motor**

- Lift Weight = $-128.08x^2 + 419.75x + 37.647$
- $R^2 = 0.9749$

**Himax motor**

- Lift Weight = $-418.75x^2 + 779.69x + 10.965$
- $R^2 = 0.9915$

**Hacker motor**

- Lift Weight = $187.07x^2 + 250.87x - 15.366$
- $R^2 = 0.9931$

*Fig. 3.11 Test brushless motors*
3.2.1 Motor conclusions

With the data obtained we can discard some pieces and get some conclusions:

- Scalextric motor cannot be used in our application because its maximum lift weight (23 g) is lower than its own weight (27 g).

- DVD motor cannot be used for the exact same reason. Its maximum lift weight is 15.4 g and its own weight 21.8 g.

As we can see, the two non-related with aeromodelling motors has not given the needed results. Only the tail motor and quadRotor motor can lift some payload weight.

- With the quadRotor motor we observe that O2 and G1 blades gives nearly the exact same lift weight, as G1 weight three times more we discard using G1.

- With low torque motors, B1 blade gives the best lift weight even being smaller than O1. That is because these motors cannot give its maximum revolutions with a bigger blade.

- With powerful enough motors, as bigger the blade is, higher the lift weight would be.

- Although they have different power, we corroborate that both tested brushless motors give higher performance than brushed motors, so as a first approximation were going to use a brushless motor.

- Both Himax and Emax motors have perfect performances for our project. The differences between them are very small. The consumption is quite equal for the same lift weight values, but Himax range is wider.

After analyzing these results we decide to use the Himax motor on our Bi-Rotor and the Hacker for the Tri-Rotor. Why Himax and no Emax? Just because of delivering problems it was easier to get a pair of Emax motors, but both of them would be a good choice. The same way it seems that we could use Hacker for both of them due to the best performances, but when testing them we also had bought Himax motors.

It seems obvious that the more expensive RC motors would work better than other types, but what we pretended by this study was to proportionate information about if other non RC motors could be used in some other future application with different economics and physics requirements.
3.2.2 Available payloads

With the previous selection of motors and blades, we can compute the total net weight that each vehicle can lift. It is an important value for developing future applications.

Table 3.2 Bi-Rotor and Tri-Rotor performances

<table>
<thead>
<tr>
<th></th>
<th>TOTAL WEIGHT</th>
<th>MAXIMUM LIFT WEIGHT</th>
<th>PAYLOAD WEIGHT</th>
<th>LIFT RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi-Rotor</td>
<td>522 g</td>
<td>750 g</td>
<td>228 g</td>
<td>1.43</td>
</tr>
<tr>
<td>Tri-Rotor</td>
<td>660 g</td>
<td>1800 g</td>
<td>1140 g</td>
<td>2.72</td>
</tr>
</tbody>
</table>

Maximum lift weight has been obtained using the data acquired from tested motors and considering the number of motors. In the case of the Tri-Rotor, we assumed that the lower motor of each pair gives only the 50% lift due to aerodynamical influence of the upper one. It is a simple approximation, and for more accurate values an aerodynamical study should be performed.

For detailed data of component weights, see Annex 2.

3.3 RF Data-Link

As we have seen two operation modes were desired for the vehicle: autonomous and controlled by the user. In this section we will see the chosen communications method that allows the user to interact with the MAV to provide the navigation orders for controlled mode.

To perform the link we decided to use a commercial radiofrequency FM data system due to its easy functioning, good response and sufficient operation distances.

3.3.1 Transmitter

A transmitter is a generator of an alternating voltage. The signal is transmitted through space using a matching antenna. This signal, by itself, cannot send information except to indicate there is a radiation source and it is coming from some point. This unchanging signal at radio frequencies is the RF carrier.

In order to transmit information, some characteristic of the transmitted signal must change; this is modulation. In model radio control the types of modulation commonly used can be in the form of on-off switching (keying) of the signal (Amplitude Modulation, AM) or changing its frequency (Frequency Modulation, FM).

All control information is sent in the form of either discrete on-off of the RF amplitude in the case of AM or a step shift of frequency in the case of FM.
The most common multi-channel digital R/C systems utilize a series of pulses to define the controlling signals for each servo channel. Next figure depicts a typical train of pulses from a five-channel transmitter; the total number of pulses in the train is one more than the number of control channels. Figure 3.12 shows a common transmitter block diagram.

![Multi-channel digital RC system diagram]

In particular, the used transmitter has been the *Futaba 4EXA* (Figure 3.13). This choice was made due to some reasons, the most important were its four channels, that allow us to control two servos (ch 2-4), two brushless motors (ch 3) and leave one extra channel for future new applications. Its single price was also an important factor to take into account.

Its main characteristics are:

- **4-model memory**: It allows us to control more than one prototype without changing the configuration of the transmitter each time.
- **Trim Memory**: To save the optimal configuration for the model.
- **EPA (End Point Adjustment) on all channels**: To configure the maximum actuation point of the servos.
- **Servo Reversing on all channels**: To reverse the response of the servos.
- Possibility to set up exponential response of the servos.
- **Possibility to mix channels**: With this feature we could synchronize the movement of the servos to obtain the desired response.
- Low battery alarm.
- Transmitter NiCd and charger.
For us, the most important feature was the possibility to program and navigate through its options menus with the ease of Mode and Select keys and lock in digitally accurate settings with the Data Input Lever.

3.3.2 Receiver

Signals from the transmitter excite the receiver's input through the receiver's antenna. Receivers have means of providing frequency selectivity in order to reject signals that are in adjacent bands. This selectivity is typically in the form of resonant circuits tuned to the received frequency.

To achieve high receiver gain without oscillation due to feedback from the later amplifier stages, a superheterodyne receiver technique is used. This type receiver uses a local oscillator to generate a signal that mixes with the incoming signal to produce sum and difference frequencies between the incoming and local oscillator frequencies. Figure 3.14 shows a block diagram of a single conversion receiver. For R/C receivers, resonant crystals control the local oscillator frequency. A tuned circuit that follows the mixer selects the lower frequency output of the mixer; this is called the intermediate frequency (IF) and is typically 455 kHz.
Another important advantage of the superheterodyne receiver is improved selectivity. Since the bandwidth of a tuned circuit relates to its center frequency, use of a lower intermediate frequency provides an inherent narrow-banding proportional to the ratio of the IF to the received frequency.

One disadvantage of such a low IF is that the receiver may detect what is called the image frequency that is located exactly the intermediate frequency on the other side of the local oscillator frequency. For example, if the local oscillator is 71.545 MHz and the IF is 0.455 MHz, frequencies of 72.000 MHz and 71.090 MHz will be detected unless a selective circuit suppresses the unwanted signal. Single conversion receivers must have very selective receive frequency circuits to reduce the image frequency response.

To correct this problem dual conversion receivers are used. This concept uses two Intermediate Frequencies (IF) and two crystal controlled oscillators. The first Intermediate Frequency is higher than 455 kHz, typically 10.7 MHz. Signals that could cause spurious responses are now beyond the passband of the RF stage. A second mixer reduces the 10.7 MHz to 455 kHz to obtain a good selectivity. Due to its complexity, increased costs and added weight, such a design is not widespread among the manufactured VHF equipment, but under some severe operating conditions it may give the only solution to reliable performance.

Our receiver has been the *Futaba R136F* (Figure 3.15). This is a single conversion full range 6 channel FM receiver that works in the 35 MHz Band.

![Fig. 3.15 Futaba R136F](image)

### 3.4 Power system

As we wanted to build something quite small, weight was one of our critical design parameters. Because of that, we needed to choose the most efficient type of battery in terms of *power vs. weight*. Other factors were also taken into account to make the choice, like availability of refills, facility to recharge or price. It was also clear that what we needed was a secondary cell (rechargeable battery) to take profit of it. We studied the characteristics of the most common types of batteries used for small electrical applications as shown in Table 3.3.
Recently Zinc-Air batteries are becoming the new revolution in *Energy/Weight* terms which would make them perfect for our application, but at the moment they are produced only as primary batteries, which is not useful for us.

With these considerations, we decided to use a lithium-ion polymer battery (LiPo). This kind of battery is a rechargeable battery which has technologically evolved from lithium-ion batteries. Ultimately, the lithium-salt electrolyte is not held in an organic solvent as in the lithium-ion design, but in a solid polymer composite such as polyethylene oxide or polyacrylonitrile. The advantages of LiPo over the lithium-ion design include lower cost manufacturing and being more robust to physical damage. The main difference between commercial polymer and lithium-ion cells is that in the last ones the rigid case presses the electrodes and the separator onto each other, whereas in polymer cells this external pressure is not required because the electrode sheets and the separator sheets are laminated onto each other.

Since no metal battery cell casing is needed, the battery can be lighter and it can be specifically shaped to fit the device it will power. Because of the denser packaging without inter-cell spacing between cylindrical cells and the lack of metal casing, the energy density of LiPo batteries is over 20% higher than that of a classic Li-ion battery.

Other important parameter to choose this type of battery is that it has no memory effect: battery's loss of capacity due to partial discharge cycles, like happens in Ni-Cd and Ni-MH batteries.

The voltage of a LiPo cell varies from about 2.7 V (discharged) to about 4.23 V (fully charged), and LiPo cells have to be protected from overcharge by limiting the applied voltage to no more than 4.235 V per cell used in a series combination.

Due to their low self-discharge rates and high energy storage/weight ratio, LiPo batteries are the preferred power sources for most electric modellers today and the most efficient for us.

### Table 3.3 Battery performances comparison

<table>
<thead>
<tr>
<th>Type</th>
<th>Energy/Weight (Wh/Kg)</th>
<th>Cell Voltage</th>
<th>Cycle durability (^2) (cycles)</th>
<th>Charge time</th>
<th>Self discharge rate</th>
<th>Charge/discharge efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ni-Cd</td>
<td>48-80</td>
<td>1.2 V</td>
<td>1500</td>
<td>1 h</td>
<td>30 %</td>
<td>70-90 %</td>
</tr>
<tr>
<td>Ni-MH</td>
<td>60-120</td>
<td>1.2 V</td>
<td>300-500</td>
<td>2-4 h</td>
<td>20 %</td>
<td>65 %</td>
</tr>
<tr>
<td>Li-ion</td>
<td>110-160</td>
<td>3.6 V</td>
<td>300-500</td>
<td>2-4 h</td>
<td>10 %</td>
<td>99.9 %</td>
</tr>
<tr>
<td>Li-Po</td>
<td>130-200</td>
<td>3.7 V</td>
<td>400-500</td>
<td>1-1.5 h</td>
<td>10 %</td>
<td>99.8 %</td>
</tr>
</tbody>
</table>

\(^2\) To 80% of initial capacity.
Battery Packs (Figure 3.16) consist of one or more LiPo cells. The arrangement of these cells determines the battery pack’s voltage, capacity, and C rating. The cells can be arranged in parallel and/or series where voltage is determined by the number of cells in series, and capacity by the number of blocks in parallel. Cells in a battery should be of the same type to maintain cell characteristics.

Mixing types can cause packs to underperform or deteriorate in an unsafe manner. Having a poorly matched or misused pack can also lead to unbalanced cells (more on balancing below) which can deteriorate your battery packs and lead to further complications.

As the engines were the elements which required more energy from the battery, they were the critical factors when choosing the batteries

**Bi-Rotor:**

To provide a voltage of about 11 V we needed to use a 3 cells in serie LiPo (3S), being the nominal cell voltage of about 3.7 V. The second important parameter to choose was the capacity. To do that, we made an estimation of total vehicle consumption assuming an operating time of 3 min at maximum engine power (we will work at about 40-50 %, so the real durability will be longer) which may reach approximately the 10 minutes flight, which is the common acceptable flight time for this kind of vehicles.

<table>
<thead>
<tr>
<th>Element</th>
<th>Energy consumption (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engines</td>
<td>20 A</td>
</tr>
<tr>
<td>Servos</td>
<td>&lt;0.5 A</td>
</tr>
<tr>
<td>Lightning</td>
<td>&lt;0.5 A</td>
</tr>
<tr>
<td>MSP430 (or equivalent)</td>
<td>&lt;0.5 A</td>
</tr>
<tr>
<td>Sensors</td>
<td>&lt;0.5 A</td>
</tr>
<tr>
<td>RF Receiver</td>
<td>&lt;0.5 A</td>
</tr>
<tr>
<td>CAPACITY REQUIRED (mAh)</td>
<td>( \frac{3 \text{ min} \cdot 22 \text{ A}}{60 \text{ min/h}} \cdot 1000 = 1100 \text{ mAh} )</td>
</tr>
</tbody>
</table>
The battery capacity is based on two things. The first is the capacity of a single cell, and second, how many blocks are placed in parallel. The reason we refer to these in blocks is because in order to double the capacity of a pack, the total number of cells must be double. On a 1S pack (single cell in series), a cell placed in parallel will double the capacity. However, to double the capacity of a 2S pack, a 2S block must be placed in parallel with the existing block. For example:

- 1S2P = 2 cell (single cell voltage, twice the capacity)
- 2S1P = 2 cell (twice the voltage, single cell capacity)
- 2S3P = 6 cell (twice the voltage, three times the capacity)

So, the number of blocks in parallel can be 1 (3S1P) because we do not need more capacity.

The last selection criteria for battery pack, was its C rating. The C rating of the battery is the maximum continuous discharge rating of the lipo cells and it determines the intensity (A) that your system can pull from the pack without overheating it. Discharging batteries at higher rates will raise their temperature and can lead to combustion.

This parameter is determined by divide the required maximum intensity (A) by battery pack’s capacity (Ah). In our case, we have:

$$ C_{rating} = \frac{22A}{1,1Ah} = 20 C $$  \hspace{1cm} (3.2)

With these requirements, we choose the Figure 3.17 battery for our prototype.

**Table 3.5. VTEC lithium polymer battery specifications**

<table>
<thead>
<tr>
<th>BATTERY SPECIFICATIONS</th>
<th>VTEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Producer</td>
<td>VTEC</td>
</tr>
<tr>
<td>Weight</td>
<td>93 g</td>
</tr>
<tr>
<td>Size</td>
<td>71x36x21.5 mm</td>
</tr>
<tr>
<td>C-rate</td>
<td>25 C</td>
</tr>
<tr>
<td>Capacity</td>
<td>1100 mAh</td>
</tr>
<tr>
<td>Voltage rating</td>
<td>11.1V – 3S1P</td>
</tr>
</tbody>
</table>

*Fig. 3.17 VTEC battery*
Tri-Rotor:

With the same process and the new requirements for our second vehicle we found the characteristics of the battery for the Tri-Rotor.

In this case, the maximum voltage to provide was 7.4 V due to motors characteristics. So, the new number of LiPo cells in serie must be 2 (2S).

In this case, capacity required was bigger due to the fact that we had six engines with a maximum consumption of about 7A each. So, now, the determinant intensity was fixed on about 45 A.

Table 3.6. Tri-rotor consumptions and capacity required

<table>
<thead>
<tr>
<th>Element</th>
<th>Energy consumption (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engines</td>
<td>43 A</td>
</tr>
<tr>
<td>Lightning</td>
<td>&lt;0.5 A</td>
</tr>
<tr>
<td>MSP430 (or equivalent)</td>
<td>&lt;0.5 A</td>
</tr>
<tr>
<td>Sensors</td>
<td>&lt;0.5 A</td>
</tr>
<tr>
<td>RF Receiver</td>
<td>&lt;0.5 A</td>
</tr>
</tbody>
</table>

\[
\text{CAPACITY REQUIRED (mAh)} = \frac{3 \text{ min} \cdot 45 A}{60 \text{ min}/1 \text{ h}} \cdot 1000 = 1100 \text{ mAh}
\]

As before, was enough with a 2S1P pack, and the new minimum C rating was fixed at:

\[
c_{\text{rating}} = \frac{45 A}{2.25 A \cdot h} = 20 \text{ C}
\]  

(3.3)

So, the second battery was (Figure 3.18):

Table 3.7. VTEC lithium polymer battery specifications

<table>
<thead>
<tr>
<th>BATTERY SPECIFICATIONS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Producer</td>
<td>Thunder Power RC</td>
</tr>
<tr>
<td>Weight</td>
<td>154 g</td>
</tr>
<tr>
<td>Size</td>
<td>15x39x118 mm</td>
</tr>
<tr>
<td>C-rate</td>
<td>30 C</td>
</tr>
<tr>
<td>Capacity</td>
<td>2700 mAh</td>
</tr>
<tr>
<td>Voltage rating</td>
<td>7.4 V – 2S1P</td>
</tr>
</tbody>
</table>

Fig. 3.18 Thunder Power RC battery
Its performances are higher than needed, but we decided to acquire it due to its availability and \(\frac{\text{performance}}{\text{price}}\) ratio.

Although these batteries have been chosen, they could be changed for different requirements. For example, if a little payload had to be lifted, a bigger, heavier and with more capacity battery could be used in order to extend the flight time.

### 3.5 Microcontroller and boards

There are many microcontroller series to work with, each one with its own advantages. Any of them would be suitable for our project. In fact we started learning and programming with the MSP430 (Figure 3.19) environment but due to a lack of time we finally used an Arduino board (Figure 3.20). Let’s see why we did that change looking at the advantages of this pair of microcontrollers.

**Advantages:**
- Lowest consumption.
- Programming efficiency.

**Disadvantages:**
- Programming hardware needed.
- Design of an own PCB board needed.
- More complicated programming code.

**Advantages:**
- Already mounted on a PCB.
- On-board USB connector to program it.
- Easier to write code, due to the libraries and examples uploaded by users to Arduino’s site.

**Disadvantages:**
- Less customizable.
- Bigger size.
- Shield needed (complementary board).
Initially, we decided to use MSP430 for our project because we wanted to develop our own custom PCB board designed for our interests, but due to a lack of time we finally moved to Arduino platform to achieve some of the wanted objectives with less time. This is not a problem because the microcontroller can be changed anytime with a few changes in the code. All proposed codes from Annex 1 are developed in a MSP430 environment except the used in Tri-Rotor which are developed on the Arduino platform.

We need to control many actuators. Each one needs a specific voltage values which will drain different current values and also signal generated from Arduino. All of these means lots of wire going from the board and the battery to the actuators and all of them connected to common ground. It is easy to see that either on the Bi-Rotor (4 actuators) or the Tri-Rotor (6 actuators) this would be a mess of wires. In order to avoid that and minimize the total length of wire, we designed a personalized board for each MAV that also will allow us to easily plug-in any desired sensor.

In the case of the Bi-Rotor we called it **PowerShield**, and for the Tri-Rotor we called it **FlyShield**.

### 3.5.1 FlyShield

Including the access to all of the Arduino’s pins, **FlyShield** (Figure 3.21) also provides:

- Six commercial Servo / Brushless motor connector (signal + power + ground).
- Integrated accelerometer with a low-pass filter and an amplifier.
- Integrated gyroscope with a band-pass filter and an amplifier.
- Six power connectors with the voltage of the connected battery for the motors.
- Powers up Arduino from the battery.

![Fig. 3.21 FlyShield](image)
Figure 3.22 shows a detailed view of the board.

3.5.1.1 Accelerometer

An accelerometer is an electronic device sensible to the changes of acceleration. Single and multi-axis models are available to detect magnitude and direction of the acceleration, and can be used to sense orientation, vibration and shock.

In our case we are going to use a 3-axis model (Figure 3.23) to determine the attitude of our MAV and try to stabilize it with its readings. Although a two-axis accelerometer would be enough for our purposes, the third axis could be used in a future for various applications.

Fig. 3.22 FlyShield detailed view

Fig. 3.23 ADXL3xx accelerometer
The accelerometer has three signal outputs (one for each axis), and the voltage value of each of these pins depends on the angle this axis forms with the horizontal. So, reading this voltage values and knowing this relationship between angle and voltage, we can determine the attitude of our MAV.

To find out this relationship we need to experimentally get this information because it is not given in the device’s datasheet. We read the output voltages for various representative angles in order to estimate the real output of our accelerometer for both X and Y axis. We do not get the Z axis information because it is not needed in our application.

### Table 3.8 Accelerometer output voltage values with $V_{cc} = 3.3 \, V$

<table>
<thead>
<tr>
<th></th>
<th>-90°</th>
<th>-75°</th>
<th>-60°</th>
<th>-45°</th>
<th>-30°</th>
<th>-15°</th>
<th>0°</th>
<th>15°</th>
<th>30°</th>
<th>45°</th>
<th>60°</th>
<th>75°</th>
<th>90°</th>
</tr>
</thead>
<tbody>
<tr>
<td>X axis</td>
<td>1.367</td>
<td>1.378</td>
<td>1.416</td>
<td>1.467</td>
<td>1.541</td>
<td>1.623</td>
<td>1.717</td>
<td>1.803</td>
<td>1.895</td>
<td>1.954</td>
<td>2.004</td>
<td>2.036</td>
<td>2.042</td>
</tr>
<tr>
<td>Y axis</td>
<td>1.380</td>
<td>1.390</td>
<td>1.422</td>
<td>1.479</td>
<td>1.549</td>
<td>1.637</td>
<td>1.714</td>
<td>1.814</td>
<td>1.896</td>
<td>1.971</td>
<td>2.026</td>
<td>2.058</td>
<td>2.072</td>
</tr>
</tbody>
</table>

With these values we can extract the following lineal approximation of the output voltage value depending on the inclination angle ($\theta$) for any axis with a correlation parameter of 0.97. It is done by using the arithmetical mean of both axes because theoretically they have the same behaviour.

$$V_{acc}(\theta) = 4.4 \cdot 10^{-3} \theta + 1.72 \quad (3.4)$$

As the accelerometer is mounted on the MAV, it senses the structural vibration produced by its actuators. In order to eliminate these readings, we use a passive low-pass filter because vibrations are of a higher frequency than the attitude changes of the vehicle.

Also, we see that the output signal limits of the accelerometer are not the input limits of the used ADC and that reduces the sensibility of the system. To prevent that and use the entire width of our ADC we amplify the output signal using an operational amplifier.

So, by filtering and amplifying as shown in Figure 3.24 [16] we improve the accelerometer signal and make it useful for our specific amplification.

![Amplification and filtering circuit](image-url)
This is the transfer function of the circuit:

\[ V_{out} = -V_{ref} \cdot \frac{R_2}{R_1} + V_{acc} \cdot (1 + \frac{R_2}{R_1}) \]  \hspace{1cm} (3.5)

We need to choose \( V_{ref} \), \( R_2 \) and \( R_1 \) for conditioning the signal as best as possible, in order to use all the width of the ADC entry.

For choosing the resistances we have to calculate the gain we need. The first thing is to decide which values are the desired minimum and maximum accelerometer outputs. In a first approximation we thought of taking the 90º and -90º voltage values (Table 3.8) and that would work quite well, but then we decided to choose 60º and -60º values as margins because the vehicle is not supposed to reach a 90º degrees inclination. This way we achieve more resolution when determining the vehicle’s attitude. So, \( V_{acc}^{\text{min}} = 1.4 \text{ V} \) and \( V_{acc}^{\text{max}} = 2.0 \text{ V} \).

Because we do not have an offset to modify, we can compute the gain in order to reach 0 V (ADC minimum entry) for the desired minimum accelerometer output or 5 V (maximum ADC entry) for the desired maximum accelerometer output. Let’s compute both of them:

From 3.5,

\[ \frac{R_2}{R_1} = \frac{V_{acc} - V_{out}}{V_{ref} - V_{acc}} \] \hspace{1cm} (3.6)

For reaching \( V_{acc} = 0 \text{ V} \) with \( V_{acc}^{\text{min}} \)

\[ \frac{R_2}{R_1} = 7.8 \] \hspace{1cm} (3.7)

For reaching \( V_{acc} = 5 \text{ V} \) with \( V_{acc}^{\text{max}} \)

\[ \frac{R_2}{R_1} = 7.1 \] \hspace{1cm} (3.8)

Because we do not need specifically any of the two situations, we chose a medium value. This way, as we will see, we get an acceptable range of output
levels centered on the zero degrees voltage output. For choosing the $R$ values we tried to get the most approximated gain value to 7.45, which is exactly the mean value of the both shown before.

We chose $R_1 = 120 \, k\Omega$ and $R_2 = 910 \, k\Omega$:

$$\frac{R_2}{R_1} = 7.6 \quad (3.9)$$

With this relationship we get the following results:

$$V_{acc}^{\text{min}} = 1.40 \, V \rightarrow V_{out}^{\text{min}} = 0.04 \, V \quad (3.10)$$

$$V_{acc}^{\text{max}} = 2.00 \, V \rightarrow V_{out}^{\text{max}} = 5.19 \, V$$

So, this will be the range of values that the ADC will receive.

### 3.5.1.2 Gyroscope

To quantify the rotation in the horizontal plane (X/Z-axis) in order to correct it we planned the use of a rotational angular velocity sensor, commonly termed gyroscope.

In our case, we used a vibrating structure gyroscope. They are solid state devices which provide an output voltage proportional to the rate of turn applied to the sensitive axis.

**Main features of Vibrating Structure Gyrosopes:**

- No moving parts.
- Simple construction.
- Short start-up time.
- Low power requirements.
- Low noise.

These vibrating structure devices work on the basic principle of detecting Coriolis forces. These forces are generated when a moving particle is rotated. To use the Coriolis effect to detect angular rotation, a solid structure is forced to vibrate normally at its resonant frequency. This is achieved by applying an alternating voltage to the primary electrodes. The vibration provides the structure with a linear velocity component. When the structure is rotated, the Coriolis forces cause the vibration motion of the structure to be coupled to
another vibration mode or plane of the structure. The magnitude of this secondary vibration is proportional to the angular rate of turn (Figure 3.25).

![Gyroscope scheme](image)

**Fig. 3.25 Gyroscope scheme**

To adapt the accelerometer output, the scheme given in figure 3.26 has been used:

![Accelerometer scheme](image)

**Fig. 3.26 Accelerometer scheme**

In the first stage, a high pass filter with an approximate cut-off frequency of 0.3 Hz is used to eliminate DC component caused by temperature drift (due to change of ambient temperature).

\[
f_c = \frac{1}{2\pi RC} \quad \rightarrow \quad f_c = \frac{1}{2 \cdot \pi \cdot 10^5 \cdot 4.7 \cdot 10^{-6}} \rightarrow f_c = 0.3 \text{ Hz} \quad (3.11)
\]

Then the signal is amplified and guided through a low pass filter to suppress output noise component around 30-33 kHz (resonant frequency of sensor element). This filter has higher cut-off frequency than required to give an extensive margin and obtain accurate results, in particular it has about 1 kHz.
\[ f_c = \frac{1}{2\pi RC} \rightarrow f_c = \frac{1}{2 \cdot \pi \cdot 9 \cdot 10^4 \cdot 1.8 \cdot 10^{-9}} \rightarrow f_c = 982.4 \text{ Hz} \] (3.12)

### 3.5.2 PowerShield

Including the access to all of the MSP430 pins, *PowerShield* (Figure 3.27) also provides:

- Two power connectors with the voltage of the connected battery for the motors.
- One power connection at 5 V for the RF receiver.
- Three LED connections.
- One light sensor connection.
- Powers up MSP from the battery at 3.3 V.

**Fig. 3.27** PowerShield

Figure 3.28 shows a detailed view of the board.

**Fig. 3.28** PowerShield detailed view
In this chapter we have seen the different parts of our MAVs designs. We have the RF-Datalink which will allow us to directly control the different actuators such as servos and motors. The microcontroller and our own boards give us the possibility to read various sensors, stabilize the vehicle and make it an autonomous vehicle. With all of that, what is missing is to develop software that provides the vehicle with the desired functionalities.
CHAPTER 4. SOFTWARE

4.1 AVOIDANCE SYSTEM

4.1.1 Objective

The objective of this system is to give the MAV the capacity of detecting and avoiding collisions with objects or walls. This is done using some distance sensors its voltage changes in function of the distance of the object in front of it. Using this information, we will be able to deflect the motors consequently to avoid the collision.

With this same system the vehicle can realize automatic “soft” landings using the information of the ground distance sensor and also using the accelerometer data.

Shown in Figure 4.1 are the 4 distance sensors situated on the Bi-Rotor structure. We have decided to use four of these sensors in order to detect walls, the roof and the floor (the last one especially useful to achieve soft landings).

![Distance sensors localization](image)

**Fig. 4.1.** Distance sensors localization

Also, using the sensor 4 information we can reduce the impact on an automatic landing, preventing the structure to get damaged. This system would read the floor distance sensor and act on both motors in order to perform a smooth approximation to the ground.
4.1.2 Parts

This system involves the following parts of the MAV:

- 4 distance sensors
- Both motors
- Both servos
- The microcontroller

4.1.3 Code

Using any timer of the MSP430 we can generate a pulse signal as needed. This is done using the compare mode, which basically puts the output on a logical high when the timer reaches a determined value (which we can vary on the code) and returns the output to a logical low when something else happens (depending on the output mode selected this event can be the return to 0, when another value is reached, and more).

For controlling the servos we are going to use the output mode 3 which toggle the output into a high state each time the timer reaches the TACCRx value (the \( x \) is the capture/compare port number, depending on the microcontroller there will be more or less of these outputs) and puts it off when it reaches TACCR0, with the timer set in the “Up” mode (which counts from 0 to TACCR0 repeatedly). Figure 4.2 represents it.

![Fig. 4.2 Generation of a PWM using MSP430](image)

Looking at the picture, it is easy to see that TACCR0 is related with the period signal \( T \) and TACCRX with the duty cycle \( D_c \) of the PWM, so varying these two parameters we will be able to choose the signal frequency and the duty cycle which will allow us to control the servo’s position. What we have to do now is see which values of TACCR0 and TACCRx do we have to use in order to achieve the times needed.
These are the two formulas to calculate our TACCR0 and TACCRx values, where:

✓ \( f_{\text{source of the timer}} \) is the frequency of the clock which is using the timer.

✓ \( \text{pre - scaler} \) is a parameter selectable from the timer registers which allows us to divide the frequency clock by some powers of two.

✓ \( f_{\text{desired}} \) is the frequency signal that we want.

**IMPORTANT:** On most of the small microcontrollers there are 8-bit and 16-bit timers, this means the first ones counts from 0 to 256 \((2^8)\) and the second ones from 0 to 65536 \((2^{16})\). You need to verify that your desired TACCR0 is below this numbers, if not, you will have to increase the pre-scaler value until TACCR0 falls between the timer extreme values.

Using these general formulas with the parameters we need for controlling the servos we get the following results.

\[
TACCR0 = 19999
\]

\[
TACCRx_{\text{extreme 1}} = 1999.9 \quad TACCRx_{\text{centered}} = 1499.9 \quad TACCRx_{\text{extreme 2}} = 999.9
\]

As we see, we have 1000 different selectable values between both extremes of the servo position. There is no servo that has such number of physical positions, so lots of them will be duplicated. It means that if we change TACCRx from 1607 to 1608 the position of the servo may not change. If we have the total number of different positions of our servo, we will be easily able to divide this 1000 different TACCRx values by the real number of positions in order to get how many values we have to increment/decrement TACCRx to get the minimum position change.

The next step is to program a software in order to read the distance sensors information and act as we want. The proposed code can be found in *Annex 1* divided in functions.
4.2 GROUND CONTACT SYSTEM

4.2.1 Objective

The main objective of this system is to provide the MAV the capacity of recognizing if it is landed on the ground or not. This is done with one microswitch in each leg of the vehicle, which will be turned on when it contacts the ground.

With this information available we can think many code applications to control the behaviour of the MAV, such as automatically power-off the engines on automatic mode, act on different lights or parts of the vehicle, and more.

4.2.2 Parts

This system involves the following parts of the MAV:

- 3 microswitches (one on each leg).
- The microcontroller.

4.2.2.1 Microswitches

A microswitch is basically one push-button which is activated with a low pressure. It has three pins: a common one and the other pair connected to one of the two possibilities depending on the state of the button.

In our case we just want to differentiate if the button is “on” or “off” using the microcontroller, so the variable pins of the microswitch will be \( V_{cc} \) and Ground and the common pin will be connected at the microcontroller input port.

![Microswitch connection](image-url)

Fig. 4.3. Microswitch connection

There are different ways to connect the microswitch, each way with its own results. We decided to connect them as shown in Figure 4.3 because the other connections had disadvantages like a continuous consume and the need to use some resistors.
4.3 LIGHTNING SYSTEM

4.3.1 Objective

The aim of this system is to automatically turn the lightning system on when the “outside” light decreases under a determined threshold using a light sensor.

This is done by LEDs distributed on the MAV structure as follows (Figure 4.4). There is one flashing ultra-bright LED under each motor, one red and the other one green, which we are going to call “Motor LEDs”. One white low-intensity LED in each floor level of the structure in order to recognize better de μAV into the darkness and be able to operate on its electronics without an external light. We are going to refer them as “Body LEDs”. And finally, an ultra-bright white LED facing the ground called “Landing LED” which will only be activated when the global lightning system is activated and ground proximity is detected using a distance sensor explained on the Avoidance System chapter.

4.3.2 Parts

This system involves the following parts of the MAV:

- A light intensity detector.
- 2 “Motor LEDs”, one on each motor.
- 2 “Body LED2” one on each floor level.
- 1 “Landing LED” facing the ground.
- The microcontroller.

Let’s see with more profundity each of these items, how they work, important values and parameters and how we reached them.

4.3.2.1 LEDs

They are the main part of the system. We need to compute which resistance add to each LED (Figure 4.5) in order to control the consumption and reach the desired brightness. This is done as follows:
Fig. 4.5. Led circuit scheme

\[ R = \frac{V_s - V_{LED}}{I} \]  \hspace{1cm} (4.3)

With this expression we can find the resistance value needed by measuring \( V_{LED} \) with a multimeter and select the desired current (which will vary the luminosity). See Figure 4.6 for detailed values.

4.3.2.2 Light intensity sensor

For automatically turning on the lights when it is needed we use a Light Dependent Resistor (LDR) (Figure 4.7), which is a resistor that changes its resistance depending on the light intensity.

Fig. 4.6. Various LED parameters

- **V\text{BLUE LED} = 2.8 V**
  - \( R\text{BLUE LED} = 25 \Omega \)
  - \( I\text{BLUE LED} = 5 \text{ mA} \)

- **V\text{WHITE LED} = 2.8 V**
  - \( R\text{WHITE LED} = 100 \Omega \)
  - \( I\text{WHITE LED} = 2 \text{ mA} \)

- **V\text{RED LED} = 2.1 V**
  - \( R\text{RED LED} = 100 \Omega \)
  - \( I\text{RED LED} = 9 \text{ mA} \)

- **V\text{GREEN LED} = 2.7 V**
  - \( R\text{GREEN LED} = 25 \Omega \)
  - \( I\text{GREEN LED} = 3 \text{ mA} \)
We had no datasheet of the photoresistor we have used, so after thinking about the circuit we need to condition the sensor we have to determine some interesting values.

\[ R_{\text{max}} \gg \text{Total darkness} \gg 28 \, \text{k}\Omega \]
\[ R_{\text{min}} \gg \text{Total light} \gg 1.2 \, \text{k}\Omega \]

Now that we have this data we can move on to the next step. We need to convert this change in resistance to a voltage change because the microcontroller cannot read resistance values; we have to give him voltages. The circuit to do that is quite simple: a tension divisor where one of the resistors is our photoresistor.

The value of \( R \) must be quite big in order to reduce the consumption of the circuit, but not so big because the bigger, the smaller voltage will fall on the sensor, so we have to decide taking this compromise in account (Figure 4.8).

\[ V = R_{\text{min}} \cdot I_{\text{max}} \rightarrow 5 \, V = 10 \, k\Omega \cdot I_{\text{max}} \rightarrow I_{\text{max}} = 0.5 \, mA \] (4.4)

\[ V_{R_x} = 5 \, V \cdot \frac{R_x}{R_x + R} \rightarrow V_{R_x} = 5 \, V \cdot \frac{1.2 \, k\Omega}{11.2 \, k\Omega} \rightarrow V_{R_x} = 0.54 \, V \] (Total Light) (4.5)

\[ V_{R_x} = 5 \, V \cdot \frac{R_x}{R_x + R} \rightarrow V_{R_x} = 5 \, V \cdot \frac{28 \, k\Omega}{38 \, k\Omega} \rightarrow V_{R_x} = 3.68 \, V \] (Total Darkness) (4.6)
With this, we got the sensor output in voltage and now we are able to determine the threshold. (Note that the decision of this parameter is based on try and error method, this means we have tried different light intensities and decided when we want to turn on the lights).

\[
\begin{align*}
V_{Rx} &= 0.45 \text{ V} \\
V_{Rx} &= 1.81 \text{ V} \\
V_{Rx} &= 2.5 \text{ V}
\end{align*}
\]

Note that the LDR and the LEDs are facing opposite directions, so the activation of the lightning system will not affect at LDR readings. Otherwise this will cause an oscillation effect on turning ON and OFF the lightning system continuously.

### 4.3.3 Code

As we know, to make this system operative, we need to read the voltage value of our sensor and act on the LEDs consequently.

#### 4.3.3.1 Reading the light intensity sensor

The first step is done using the internal 10-bits ADC of the MSP, called ADC10. It has many analog inputs, the light intensity sensor is connected to A0 (pin 8). The upper and lower power limits of this ADC can be modified using two programmable/selectable voltage levels via software. It gives us the freedom of choosing the limits which better adapt to our requirements. We have different options like using the microcontroller alimentation, some pre-defined voltage level, or even external voltages.

We decide to use the main alimentation to establish these limits because we already know that our sensor will work between 0.45 V and 2.5 V and the main alimentation is 3 V. It means we have an ADC with a range from 0 to 3 V, where the 0 V will be a binary 0 and the 3 V will be a binary 1024 (10 bits). Note that a better resolution could be reached setting the input limit values of the ADC equal to the output limit values of the sensor, but it is not necessary because of the criticality level of the system.

This ideal behaviour of an ideal ADC is explained on Figure 4.9 using a 3-bits ADC in order to simplify, but it can be extrapolated to whatever number of bits.
We already established which voltage value will be our light intensity threshold, now with this formula we can find the corresponding output of the ADC10.

\[
N_{ADC} = 1023 \cdot \frac{V_{IN} - V_{R_-}}{V_{R_+} - V_{R_-}} \quad \rightarrow \quad N_{ADC} = 1023 \cdot \frac{V_{threshold} - V_{SS}}{V_{CC} - V_{SS}} \quad \rightarrow \quad N_{ADC} = 620
\]  

(4.7)

So this is the \(N_{ADC}\) threshold value which will decide if turn on the system or not.

### 4.3.3.2 Acting on the LEDs

LEDs distributed on the vehicle are connected to the following microcontroller general purpose I/O pins:

- Motor LEDs \(\rightarrow\) Pin 26 (P3.5)
- Body LEDs \(\rightarrow\) Pin 25 (P3.4)
- Landing LED \(\rightarrow\) Pin 14 (P3.3)

In order to toggle them we just need to turn each of these pins from a logical high to a low state or vice versa.

The body LEDs are the simplest because they are steady, so we just need to switch based on the digital output of the ADC10. The landing LED is exactly the same but also the ground sensor must advise of ground proximity to turn it on.

The motor LEDs are flashing and we need to use a timer and an interrupt service routine in order to create this flashing while the microcontroller is still available for the rest of the program. As shown in the code (Annex 1), this is done using de 16-bits “Timer A” from our MSP430. This timer has a list of software-configurable parameters that we can adjust based on our requirements, such as the clock source select, the counting mode and more. Then, using the “compare” option we set a point (TACCR0) between 0 and \(2^{16}\) where the interrupt will be executed when the timer reaches it.
CHAPTER 5. CONCLUSIONS AND FUTURE PERSPECTIVES

During last years studies and research of Micro Aerial Vehicles have been taking importance between some of the most important technological development groups of the world. Their wide range of applications from military missions to rescue ones has produced an increasing demand of them. Nevertheless, there are lots of unknown aspects and non-investigated fields.

In this project we have made an exhaustive research and development of this kind of vehicles accomplishing most of our previous objectives. We have understood basic MAVs theory dividing our main task into sub problems and formulating measurable demands and specifications. Two new prototypes have been totally designed and created applying the previous acquired knowledge about this field, making an analysis of the main magnitudes involved on its maneuvers that justify the particular chosen design on each case.

Mechanical phase has been an important part of this project, where materials has been chosen, pieces created, everything assembled and emergent physic problems solved.

Hardware and software implementation has been also two important phases of our work. Besides structural designs, some systems have been also developed (power, lightning, ground contact) or applied (RF data-link). Sensor data collection has meant an unavoidable task in order to get various sub-systems to interact correctly.

During the project some tests and verifications have been made in order to show that our choices and decisions provide us an efficient way to accomplish the previously fixed requirements.

It is worth noticing that even that the project aims for a complete MAV studies and development, many of the goals of it are not finished or are considered sub-optimal solutions. This project is considered as the first step and approximation to the study of Bi-Rotor and Tri-Rotor vertical take-off and landing MAVs which has produced two robust platforms that could help researchers in further investigations, leaving the optimization of each component or development of new ones to later design iterations.

As future perspectives we expect that further studies could provide our platforms of new sub-systems, like fully self-stabilization, GPS or video cameras, which correctly integrated with the provided systems, could make possible an autonomous self controlled pre-programmed flight to make our created prototypes useless for commercial applications like industrial surveillance or processes monitoring. It has been also studied the possibility of provide the vehicles with an auto-charging autonomous system that allow them to detect when their battery level reaches critical values and, with a pre-programmed flight path, order and guide the vehicle to an auto-charging dock installed in some point of the working environment.
REFERENCES


ANNEX 1. MSP430 SOFTWARE

Here are the proposed codes for each system working with an MSP430. All of these codes are thought for Bi-Rotor MAV but can be easily readapted to any other MAV by resituating the sensors and adapting the code to the vehicle’s geometry.

AVOIDANCE SYSTEM

The code for this system is divided into two parts that can be programmed in functions to make the code simpler and “friendlier”:

✓ **Reading the distance sensor**: How to read an analog port and work with the value obtained.

✓ **Avoidance**: Decisions of the behaviour of the vehicle in function of the previous readings.

**Reading the distance sensor**

In order to get the data from the distance sensors and use it for different purposes, we define a function called “distance” which will return the read distance divided into the seven levels of actuations we said before.

\[ T_{ACC R_x_{extreme}} = 1999.9 \]
\[ T_{ACC R_x_{centered}} = 1499.9 \]

So, dividing the 500 possible variation of TACCRx by 6 steps we get that we have to vary the TACCRx value 83.33 for each step.

Let’s see the code:

```c
#include "msp430x22x4.h" // Library for the specific device

//////////////////////////////////
// Returns a value depending on the distance given by the sensor
// 0 - No deflection of motors needed
// 1 - First step of deflection
// 2 - Second step of deflection
// ...
// 6 - Maximum deflection
//////////////////////////////////

int distance (int x)
{
    if(x==1) //Wall_distance_1
    {
        ADCCTL1 |= INCH_0                // Analog Input A0 (Wall_distance_1) selected
        ADC10CTL0 |= ENC + ADC10SC;        // Sampling and conversion start
        __bis_SR_register(CPUOFF + GIE);   // LPM0, ADC10_ISR will force exit
        return (ADC10MEM/146);            // Dividing 1024 (10 bits) in 7 steps of ~146 bits
    }
    if(x==2) //Wall_distance_2
    {
        ADC10CTL1 |= INCH_1;               // Analog Input A1 (Wall_distance_2) selected
        ADC10CTL0 |= ENC + ADC10SC;       // Sampling and conversion start
    }
}
```

---

**Adapted Code**

```c
#include "msp430x22x4.h" // Library for the specific device

//////////////////////////////////
// Returns a value depending on the distance given by the sensor
// 0 - No deflection of motors needed
// 1 - First step of deflection
// 2 - Second step of deflection
// ...
// 6 - Maximum deflection
//////////////////////////////////

int distance (int x)
{
    if(x==1) //Wall_distance_1
    {
        ADCCTL1 |= INCH_0                // Analog Input A0 (Wall_distance_1) selected
        ADC10CTL0 |= ENC + ADC10SC;        // Sampling and conversion start
        __bis_SR_register(CPUOFF + GIE);   // LPM0, ADC10_ISR will force exit
        return (ADC10MEM/146);            // Dividing 1024 (10 bits) in 7 steps of ~146 bits
    }
    if(x==2) //Wall_distance_2
    {
        ADC10CTL1 |= INCH_1;               // Analog Input A1 (Wall_distance_2) selected
        ADC10CTL0 |= ENC + ADC10SC;       // Sampling and conversion start
    }
}
```
Collision avoidance

As we said on avoidance system chapter, we decided to establish 6 different actions for the wall avoidance system, meaning by actions different actuation on the servos. This means that we are going to deflect the motors more or less depending on the distance given by the sensors. The advantage of this is that, once detected the proximity of the wall, if it continues approaching the servos will deflect more and vice versa, reaching a smooth behavior.

Let’s see the code:

```c
#include "msp430x22x4.h" // Library for the specific device

// Defining the different sensors
#define wall_distance_1 0x01 // Pin 2.3 (A3)
#define wall_distance_2 0x02 // Pin 2.4 (A4)
#define roof_distance 0x04 // Pin 3.6 (A6)
#define ground_distance 0x08 // Pin 3.7 (A7)

// Defining the different actuators
#define servo_1 0x04 // Pin 4.4 (TB1)
#define servo_2 0x08 // Pin 4.5 (TB2)
#define motor_1 0x04 // Pin 1.2 (TA1)
#define motor_1 0x40 // Pin 1.6 (TA1)

int main( void )
{
    // Stop watchdog timer to prevent time out reset
    WDTCTL = WDTPW + WDTHOLD;

    // Setting pins
    P4DIR = servo_1|servo_2; // Set Servo's pins as Outputs
    P4SEL = servo_1|servo_2; // TA Output for the Servos
    ADC10AE0 |= 0x08; // Enables pin 2.3 (A3), pin 2.4 (A4), pin 3.6 (A6) and pin 3.7 (A7) as analog inputs.

    // Setting ADC10
    ADC10CTL0 = SREF_1 + ADC10SHT_2 + REFON + ADC10ON + ADC10IE;
    }
```
/** Collision Avoidance System **/

```c
void collision_avoidance(void)
{
    if(distance(wall_distance_1)>0) // Checking if there is an alarm on Wall_1
    {
        TACCR1 = 1499 - 83*(distance(wall_distance_1)); // Act on both servos to avoid the wall
        TACCR2 = TACCR1;
    }
    if(distance(wall_distance_2)>0) // Checking if there is an alarm on Wall_2
    {
        TACCR1 = 1499 + 83*(distance(wall_distance_2)); // Act on both servos to avoid the wall
        TACCR2 = TACCR1;
    }
    if(distance(roof_distance)>0) //Checking if there is an alarm on Roof
    {
        TBCCR1 = TBCCR1 - 75*(distance(roof_distance));  // The 75 is choosen experimentally
        //TBCCRy = TBCCRx only necessary if we want to give different power to the engines
    }
    if(distance(ground_distance)>0) //Checking if there is an alarm on Ground
    {
        TBCCR1 = TBCCR1 + 75*(distance(ground_distance));
        //TBCCRy = TBCCRx only necessary if we want to give different power to the engines
    }
}
```

**LIGHTNING SYSTEM**

This software is divided into two parts:

- **Reading the light intensity sensor**: An analog sample of the sensor is obtained using the ADC and treated later.
- **Timer configuration**: TimerA is settled up for blinking the LED's as desired.

**Reading the light intensity sensor**

The function “`status()`” reads the light intensity sensor and returns a Boolean parameter with the information of the lightning system (‘ON/OFF’) that will be used in function “`lightning_system()`” which will turn ON or OFF the lights.

Let's see the code:

```c
#include "msp430x22x4.h"
#define on 1
#define off 0
volatile unsigned int j=0;

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop WDT
    //Setting ADC10
    ADC10CTL0 = ADC10SHT_2 + ADC10ON + ADC10IE; //ADC10 sample and hold time = 8xADC10CLK's
    //ADC10 on //Interrupts enabled
    //Reading the light intensity sensor
    //The function "status()" reads the light intensity sensor and returns a Boolean parameter
    //with the information of the lightning system (‘ON/OFF’) that will be used in function
    //"lightning_system()" which will turn ON or OFF the lights.
    //Let's see the code:
```

```c
#include "msp430x22x4.h"
#define on 1
#define off 0
volatile unsigned int j=0;

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop WDT
    //Setting ADC10
    ADC10CTL0 = ADC10SHT_2 + ADC10ON + ADC10IE; //ADC10 sample and hold time = 8xADC10CLK's
    //ADC10 on //Interrupts enabled
    //Reading the light intensity sensor
    //The function "status()" reads the light intensity sensor and returns a Boolean parameter
    //with the information of the lightning system (‘ON/OFF’) that will be used in function
    //"lightning_system()" which will turn ON or OFF the lights.
    //Let's see the code:
```
ADC10AE0 = 0x01;  // Enables pin 8 as an analog input (A0)

int status ()
{
   // Using ADC10
   while(1)
   {
      ADC10CTL1 |= INCH_0;  // Analog Input A0 selected (Default)
      ADC10CTL0 |= ENC + ADC10SC;  // Sampling and conversion start
      __bis_SR_register(CPUOFF + GIE);  // LowPowerMode0 // ADC10_ISR will force exit
      if(ADC10MEM > 620)  // Is the analog input higher than the threshold?
      {
         j++;  // Increments the counter
         if(j == 100)  // Analog input higher than the threshold for more than 100 ADC samples?
         {
            lightning_system(on);  // turn the lightning system on
         }
      }
      else  // If the value is lower than the threshold
      {
         lightning_system(off);  // turn the lightning system off
         j = 0;  // Reinitialize the counter
      }
   }
}

// Lightning system function (turns on or off the system based on the received value of x)
int lightning_system (int x)
{
   if(x==1)
   {
      TACCTL0 = CCIE;  // TACCR0 interrupt enabled
      P3OUT |= (bodyLEDs);  // Turn on the body LEDs
      lightning_system_status = on;  // Actualize the status of the lightning system to “on”
   }
   else
   {
      TACCTL0 &= ~CCIE;  // TACCR0 interrupt disabled
      P3OUT &= ~(bodyLEDs);  // Turn off the body LEDs
      lightning_system_status = off;  // Change the status of the lightning system to “off”
   }
}

// ADC10 interrupt service routine
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void)
{
   __bic_SR_register_on_exit(CPUOFF);  // Clear CPUOFF bit from 0(SR)
}

Timer configuration

Needed parameters of the timer are settled up for reaching the desired blinking on the LEDs.

This code has no functions because, once settled the timer, all works via hardware until an interruption from the timer is requested. Then, the code on the interrupt service routine is executed.
#include "msp430x22x4.h"

#define motorLEDs 0x20       // Pin X.5
#define bodyLEDs 0x10        // Pin X.4
#define landingLED 0x08     // Pin X.3

volatile unsigned int k=0, lightning_system_status=0; //lightning_System_status is used from ground_distance in //order to activate or not the landing LED

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD;  // Stop WDT

    //Setting the pins
    P3DIR |= motorLEDs|bodyLEDs|landingLED;        // Set LEDs' pins as outputs
    P3OUT &= ~(motorLEDs|bodyLEDs|landingLED);        // Start with the LEDs off

    //Setting Timer A (for flashing the motor LEDs)
    TACTL = TASSEL_2 + MC_2 + ID_3;  //SMCLK //Continuous Mode //Input divider by 8
    TACCR0 = 50000; //Sets the interrupt point value
    __bis_SR_register(GIE);    //General interrupt enabled

    while(1)
    {
        
    }
}

// Timer A0 interrupt service routine
#pragma vector=TIMERA0_VECTOR
__interrupt void Timer_A (void)
{
    k++;
    if(k>5)
    {
        P3OUT ^= motorLEDs;  // Toggle P1.0
        k = 0;              //Reinitialize the counter
    }
    TACCR0 += 50000;  // Add Offset to TACCR0
}
ANNEX 2. WEIGHTS

Detailed tables with weight and information of the main parts of each vehicle.

Bi-Rotor:

<table>
<thead>
<tr>
<th>PART</th>
<th>WEIGHT</th>
<th>QUANTITY</th>
<th>TOTAL WEIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steel threaded rod</td>
<td>40.3 g/m</td>
<td>x0.9 m</td>
<td>36.3 g</td>
</tr>
<tr>
<td>5 mm carbon rod</td>
<td>42.6 g/m</td>
<td>x0.24 m</td>
<td>10.2 g</td>
</tr>
<tr>
<td>2 mm carbon rod</td>
<td>4.2 g/m</td>
<td>x0.8 m</td>
<td>3.4 g</td>
</tr>
<tr>
<td>Battery</td>
<td>92.2 g</td>
<td>x1</td>
<td>92.2 g</td>
</tr>
<tr>
<td>Receiver</td>
<td>27.1 g</td>
<td>x1</td>
<td>27.1 g</td>
</tr>
<tr>
<td>Motor</td>
<td>54.0 g</td>
<td>x2</td>
<td>108.0 g</td>
</tr>
<tr>
<td>ESC</td>
<td>19.0 g</td>
<td>x2</td>
<td>38.0 g</td>
</tr>
<tr>
<td>PowerShield</td>
<td>28.3 g</td>
<td>x1</td>
<td>28.3 g</td>
</tr>
<tr>
<td>APC 8’ propeller</td>
<td>7.3 g</td>
<td>x2</td>
<td>14.6 g</td>
</tr>
<tr>
<td>Propeller attachment</td>
<td>4.4 g</td>
<td>x2</td>
<td>8.8 g</td>
</tr>
<tr>
<td>Servo</td>
<td>8.5 g</td>
<td>x2</td>
<td>17.0 g</td>
</tr>
<tr>
<td>Structure</td>
<td>70.0 g</td>
<td>x1</td>
<td>70.0 g</td>
</tr>
<tr>
<td>Additional wire</td>
<td>2.3 g/m</td>
<td>x4.3 m</td>
<td>10.0 g</td>
</tr>
<tr>
<td>Connectors</td>
<td>0.4 g</td>
<td>x7</td>
<td>2.8 g</td>
</tr>
<tr>
<td>Others</td>
<td>55.3 g</td>
<td>x1</td>
<td>55.3 g</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td><strong>522.0 g</strong></td>
</tr>
</tbody>
</table>

Tri-Rotor:

<table>
<thead>
<tr>
<th>PART</th>
<th>WEIGHT</th>
<th>QUANTITY</th>
<th>TOTAL WEIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steel tube</td>
<td>97.3 g/m</td>
<td>x0.09 m</td>
<td>8.8 g</td>
</tr>
<tr>
<td>5 mm carbon rod</td>
<td>42.6 g/m</td>
<td>x0.51 m</td>
<td>21.7 g</td>
</tr>
<tr>
<td>Battery</td>
<td>156.0 g</td>
<td>x1</td>
<td>156.0 g</td>
</tr>
<tr>
<td>Motor</td>
<td>32.9 g</td>
<td>x6</td>
<td>197.4 g</td>
</tr>
<tr>
<td>ESC</td>
<td>9.5 g</td>
<td>x6</td>
<td>57.0 g</td>
</tr>
<tr>
<td>FlyShield</td>
<td>41.0 g</td>
<td>x1</td>
<td>41.0 g</td>
</tr>
<tr>
<td>Arduino</td>
<td>27.0 g</td>
<td>x1</td>
<td>27.0 g</td>
</tr>
<tr>
<td>APC 8’ propeller</td>
<td>7.3 g</td>
<td>x6</td>
<td>43.8 g</td>
</tr>
<tr>
<td>Propeller attachment</td>
<td>8.0 g</td>
<td>x6</td>
<td>48.0 g</td>
</tr>
<tr>
<td>Structure</td>
<td>34.0 g</td>
<td>x1</td>
<td>38.3 g</td>
</tr>
<tr>
<td>Others</td>
<td>19.0 g</td>
<td>x1</td>
<td>21.0 g</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td><strong>660.0 g</strong></td>
</tr>
</tbody>
</table>
ANNEX 3. PLANS